NECIU ILEANA

CULEGERE DE PROBLEME REZOLVATE C/C++

ISBN 978-606-577-575-6

Editura Sfântul Ierarh Nicolae
2011

Recomandare

Cartea cu titlul “Culegere de probleme rezolvate C/C++” pune în evidență importanța pregătirii practice a elevilor din punct de vedere informatic. La finalizarea cursurilor liceale, elevii au de întocmit şi susținut o lucrare de specialitate, lucrare care să scoată în evidență însuşirea şi aplicarea tuturor noțiunilor teoretice şi practice de informatică acumulate de-a lungul celor 4 ani şcolari. Totodată, pentru a promova proba digitală din cadrul examenului de bacalaureat, elevii îşi dezvoltă,în cadrul orelor de informatică, TIC, ş.a., competențe digitale. Toate aptitudinile şi abilitățile de conducere a calculatorului deprinse de elevi se formează treptat, în timp, în urma unei activități consecvente.

Culegerea este bine structurată şi aranjată, astfel încât, parcurgând toate paginile, elevii să găsească un sprijin real în rezolvarea tuturor problemelor informatice care pot apare în viața liceală de zi cu zi şi nu numai.

Tocmai de aceea, autoarea şi-a propus ca, prin cele 110 probleme enunțate şi rezolvate, elevii să aibă o viziune de ansamblu asupra limbajelor de programare.

Este o culegere utilă pentru elevii claselor de matematică-informatică, de aceea recomand cartea pentru publicare.

18.05.2011

Colegiul Tehnic “Anghel Saligny” Roşiorii de Vede

prof.gr.I dr.ing. Costache Florentina-Flori

Prefață

Culegerea cuprinde probleme rezolvate în limbajul de programare C/C++ , probleme care au fost propuse la proba practică pentru examenul de atestat profesional la informatică în județul Teleorman, de a lungul anilor 2006-2011 şi se adresează tuturor elevilor de liceu, de la clasele de matematică-informatică.

Culegerea se poate dovedi utilă atât în pregatirea de zi cu zi la informatică, cât şi în pregătirea pentru examenele de atestat profesional sau de bacalaureat.

Toate programele din carte au fost verificate cu ajutorul mediului de programare Borland C.

Îmi exprim speranța ca această lucrare să fie de un real folos celor interesați.

Autoarea

ATESTAT 2006 TELEORMAN
SUBIECTE C++

1. **Să se calculeze:**

S= 1-1*2+1*2*3- … $(-1)^{n+1}$ *1*2*…*n

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
main()
{int s,p,i,n;
cout<<"introduce o valoare pentru n=";
cin>>n;
s=0;p=1;
for(i=1;i<=n;i++)
{p=p*i;
s=s+pow((-1),i+1)*p;}
cout<<"suma este "<<s;
getch();}
```

2. **Să se calculeze:**

P=1*1*2*3*5*8…*a_n unde a_n este al n-lea termen al şirului Fibonacci

```
#include <iostream.h>
#include <conio.h>
main()
{int p,n,a[100];
p=1;a[1]=1;a[2]=1;
cout<<"introduceti o valoare pentru n=";cin>>n;
for(int i=3;i<=n;i++)
{a[i]=a[i-2]+a[i-1];
p=p*a[i];}
cout<<"produsul este P="<<p;

getch();}
```

3. **Să se afişeze următoarele „piramide”, construite în funcție de numărul întreg n citit de la tastatură**

a. n n-1 n-2 n-3……3 2 1
…….
3 2 1
2 1
1

```
#include<iostream.h>
 #include<conio.h>
  main()
  {int i,j,n;
  cout<<"introdu o valoare pentru n=";
  cin>>n;
  for(i=n;i>=1;i--)
   {for(j=i;j>=1;j--)
   cout<<j;
   cout<<endl; }
 getch();}
```

b.

```
                1
              1 2 3

          ……1 2 3 4 5……
          1 2 3 4 ……(2n-1)
```

```
#include<iostream.h>
#include<conio.h>
main()
{int i,j,n,k;
cout<<"introdu o valoare pentru n=";
cin>>n;
for(i=n;i>=1;i--)
{ for(k=1;k<=i;k++)
cout<<"  ";
for(j=1;j<=2*(n-i)-1;j++)
cout<<j<<" ";
cout<<endl;}
getch();}
```

c.

```
1
1 2
1 2 3
……
1 2 3 …….n
1 2 3 …….n
 ……
1 2 3
1 2
1
```

```
#include<iostream.h>
#include<conio.h>
main()
{int i,j,n;
cout<<"introdu o valoare pentru n=";
cin>>n;
for(i=1;i<=n;i++)
 {for(j=1;j<=i;j++)
 cout<<j;
 cout<<endl;}
for(i=n;i>=1;i--)
{for(j=1;j<=i;j++)
cout<<j;
cout<<endl;}
 getch();}
```

4. Se citesc numere naturale până la introducerea unui număr par. Să se calculeze suma S a tuturor numerelor citite precum şi câtul şi restul împărțirii lui S la suma cifrelor lui S .

```
 #include<iostream.h>
#include<conio.h>
main()
{int x,y,z,s,s1,n;
s=0;
s1=0;
cout<<"introdu o valoare ";
cin>>x;
while(x % 2!=0)
{s=s+x;
cout<<"introdu o valoare ";
cin>>x;}
cout<<"suma este="<<s<<endl;
n=s ;
while(n!=0)
{s1=s1+n%10;
 n=n/10;}
 y=s/s1;
 z=s%s1;
 cout<<"citul este="<<y<<endl;
 cout<<"restul este="<<z<<endl;
 getch();}
```

5. Se citesc numere naturale până la întâlnirea numărului 12. Să se afişeze toate tripletele de numere citite consecutiv în care al treilea număr este restul împărțirii primului la al doilea.

```
 #include<iostream.h>
#include<conio.h>
main()
```

```
{int x,y,z;
 cout<<"o valoare pentru x="; cin>>x;
 cout<<"o valoare pentru y=";cin>>y;
 while((x!=12)&& (y!=12))
 {  cout<<"x este="<<x<<endl;
 cout<<"y este="<<y<<endl;
 cout<<"o valoare pentru z=";cin>>z;
  if(z==12) break;
  if (z==x%y) cout<<x<<" "<<y<<" "<<z<<endl;
  else cout<<"al treilea nr.nu este egal cu restul impartirii primelor doua numere"<<endl;
  x=y;
  y=z; }
  if((x==12)||(y==12) ||(z==12)) cout<<"ati introdus nr.12";
  getch();}
```

6. Se citesc numere naturale până la întâlnirea numărului 0. Să se afişeze toate perechile de numere citite consecutiv cu proprietatea că al doilea număr este egal cu suma cifrelor primului număr.

```
 #include<iostream.h>
#include<conio.h>
main()
{int x,y,s,z;
cout<<"introduceti primul numar"<<endl;
cin>>x;
while(x!=0)
{z=x; s=0;
 while(z)
 {s=s+z%10;
 z=z/10;}
 cout<<"introduceti al doilea numar"<<endl;
 cin>>y;
 if (y==0)break;
 if (y==s) cout<<x<<" "<<y<<endl;
 x=y;}
 if((x==0)||(y==0))cout<<"ati introdus zero";

 getch();}
```

7. Se citesc numere naturale până la întâlnirea numărului 0. Să se afişeze toate perechile de numere citite consecutiv, cu proprietatea că al doilea număr reprezintă numărul de apariţii ale cifrei 3 în pătratul primului.

```
 #include<iostream.h>
#include<conio.h>
main()
{int x,y,patrat,z,s;
 cout<<"introduceti o valoare pentru x";
```

```
cin>>x;
while(x!=0)
{patrat=x*x;cout<<"patratul este"<<patrat<<endl;
cout<<"introduceti o valoare pentru y";
cin>>y;
if(y==0) break;
s=0;
while(patrat)
{z=patrat%10;
 if (z==3)s=s+1;
 patrat=patrat/10;}
 if(y==s) cout<<x<<" "<<y<<endl;
 x=y;}
 if ((x==0) || (y==0)) cout<< "ati introdus valoarea zero"<<endl;
 getch();}
```

8. Fie vectorii X şi Z având fiecare n componente numere reale. Să se determine expresia e= minim(x1, yn)+minim(x2, yn-1)+…..+minim(xn, y1)

```
 #include<iostream.h>
#include<conio.h>
float minim( float x,float y)
{ float min;
if (x<y) min=x;
else min=y;
return min;}

main()
{float x[100],y[100],e;
int n;
cout<<"introduceti n=";
cin>>n;
for(int i=1;i<=n;i++)
{cout<<"x["<<i<<"]=";cin>>x[i];}
for(int i=1;i<=n;i++)
{cout<<"y["<<i<<"]=";cin>>y[i];}
e=0;
for(int i=1;i<=n;i++)
e=e+minim(x[i],y[n+1-i]);
cout<<" expresia este="<<e;
getch();}
```

9. Fie un vector X de numere întregi. Să se formeze un vector Y de numere întregi în care Y[i] să fie:

a. restul împărțirii lui X[i] la suma cifrelor lui X[i]

```
#include<iostream.h>
#include<conio.h>
main()
```

```
{float x[100];
 float y[100];
 int n,i,s,a;
 cout<<"introduceti dimensiunea vectorului n=";
 cin>>n;

 for(i=1;i<=n;i++)
 {cout<<"x["<<i<<"]=" ;
  cin>>x[i];s=0;
  a=x[i];
  while (a>0)
  {s=s+a%10;
  a=a/10;}
  y[i]=x[i]/s;}
  for(i=1;i<=n;i++)
 {cout<<"y["<<i<<"]="<<y[i];
 cout<<endl;} getch();}
```

b) Reprezentarea în baza 2 a numărului X[i]

```
 #include<iostream.h>
#include<conio.h>
main()
{float x[100];
 float y[100];
 int n,i,n2,p,a;
 cout<<"introduceti dimensiunea vectorului n=";

 cin>>n;
 for(i=1;i<=n;i++)
 {cout<<"x["<<i<<"]=" ;
  cin>>x[i];n2=0;p=1;
  a=x[i];
  while (a>0)
  {n2=n2+p*(a%2);
  a=a/2;p=p*10; }
  y[i]=n2;}
  for(i=1;i<=n;i++)
 {cout<<"y["<<i<<"]="<<y[i];

 cout<<endl;}
 getch();}
```

10. Fie vectorul X cu n componente numere întregi. Să se determine minimul componentelor negative din vector.

```
 #include<iostream.h>
#include<conio.h>
main()
{int x[100];
 int n,i,min;
```

```
cout<<"introduceti dimensiunea vectorului n=";
cin>>n;min=0;
for(i=1;i<=n;i++)
{cout<<"x["<<i<<"]=" ;
 cin>>x[i];
 if(x[i]<0)
 if(x[i]<min)
 min=x[i];}
 cout<<"minimul componentelor negative este="<<min;
 getch();}
```

11. Să se descompună un număr în factori primi memorând rezultatul sub forma unui vector de înregistrări, fiecare înregistrare va cuprinde două componente: una indicând factorul prim, iar cealaltă puterea la care apare acesta în descompunere.

```
#include<iostream.h>
#include<conio.h>
main()
{int n,i,j,k;
struct vect{int fact;
           int putere;};
vect v[100];
cout<<"introduceti numarul n=";
cin>>n;
i=2; j=1;
while(n>1)
{if(n%i==0)
{k=0;
 while(n%i==0)
 {k=k+1;n=n/i;}
v[j].fact=i;v[j].putere=k;
cout<<v[j].fact<<"la puterea ";
cout<<v[j].putere;
cout<<endl;}
i=i+1;j=j+1; }
getch();}
```

12. Matricea M cu m linii şi n coloane având componente întregi se găseşte în fişierul matrice.txt. Să se aranjeze elementele matricei astfel încât în final să fie ordonate crescător pe linii şi pe coloane. Rezultatul se va afişa în acelaşi fişier. Valorile lui m şi n nu se cunosc.

```
#include<fstream.h>
#include<conio.h>
main()
{fstream f("fis1.txt",ios::in);
char c[100];
int a[100][100],b[100][100];
```

```
int m,k,i,j,x,n=0;
while(f.get(c,100)){
 n=n+1;m=strlen(c);
f.get();}
  m=m-m/2;

cout<<"numarul de coloane este="<<m<<endl;
 cout<<"numarul de linii este="<<n<<endl;
 fstream g("fis1.txt",ios::in);
   for(i=1;i<=n;i++)
    for(j=1;j<=m;j++)
      g>>a[i][j];
g.close();
cout<<"matricea initiala este"<<endl;
 for(i=1;i<=n;i++)
 {for(j=1;j<=m;j++)
  cout<<a[i][j];
  cout<<endl;}
 for(i=1;i<=n;i++)
  for(j=1;j<=m-1;j++)
   for(k=j+1;k<=m;k++)
   if(a[i][j]>a[i][k])
    {b[i][j]=a[i][j];a[i][j]=a[i][k];a[i][k]=b[i][j];}
cout<<"matricea ordonata pe linii este"<<endl;
    for(i=1;i<=n;i++)
 {for(j=1;j<=m;j++)
  cout<<a[i][j];
  cout<<endl;}
 for(j=1;j<=m;j++)
  for(i=1;i<=n-1;i++)
   for(k=i+1;k<=n;k++)
  if(a[i][j]>a[k][j])
    {b[i][j]=a[i][j];a[i][j]=a[k][j];a[k][j]=b[i][j];}
cout<<"matricea ordonata pe linii si coloane este"<<endl;
 for(i=1;i<=n;i++)
 {for(j=1;j<=m;j++)
  cout<<a[i][j];
  cout<<endl;}

getch();}
```

13. Să se afiseze valorile şi pozitiile tuturor punctelor şa dintr-o matrice M(mxn). Punctul şa reprezintă valoarea minimă pe linie şi maximă pe coloană.

```
#include<iostream.h>
#include<conio.h>
main()
{int n,i,j,l,c,min,m,ok,a[100][100];
cout<<"introduceti dim.n=";
```

```
cin>>n;
cout<<"introduceti dim.m=";
cin>>m;
for(i=1;i<=n;i++)

 {for(j=1;j<=m;j++)cout<<"a["<<i<<","<<j<<"]=";cin>>a[i][j];};
 cout<<endl;}
 for(i=1;i<=n;i++)
 {min=a[i][1];ok=1;
 for(j=1;j<=n;j++)
 if(a[i][j]<min)
   {min=a[i][j];c=j;}
 for(l=1;l<=n;l++)
 if(a[l][c]>min) ok=0;
 if (ok)
  {cout<<"punctul sa este="<<min<<endl;
       cout<<"apare pe pozitia ["<<i<<"]"<<"["<<c<<"]";}
   }
 getch();}
```

14.Să se concateneze două fisiere text. Rezultatul concatenării va fi pus în:

a. primul fişier

```
#include<fstream.h>
#include<iomanip.h>
main()
{ fstream f ("fisier1.txt",ios::app);
  fstream g("fisier2.txt",ios::in);
  char ch;
  while(g>>resetiosflags(ios::skipws)>>ch)
  f<<ch;
  f.close();g.close();
  }
```

b. în fişier nou.

```
#include<fstream.h>
#include<iomanip.h>
main()
{ fstream f ("fisier1.txt",ios::in);
  fstream m("fisier3.txt",ios::out);
  char ch;
  while(f>>resetiosflags(ios::skipws)>>ch)
  m<<ch;
  f.close();m.close();
```

```
 fstream g("fisier2.txt",ios::in);
 fstream n("fisier3.txt",ios::app);
 while(g>>resetiosflags(ios::skipws)>>ch)
 n<<ch;
 g.close();}
```

15. Să se afişeze primele n numere prime care au suma cifrelor mai mică sau egală cu m.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
main()
{int n,m,x,i,ok,s,nr,nr1;
 cout<<"introduceti o valoare pentru m=";
 cin>>m;
 cout<<"introduceti o valoare pentru n=";
 cin>>n;
 x=0; nr=3;cout<<"1"<<endl;
 while(x<n-1)
 { i=2;
   ok=1;
   while(i<=sqrt(nr)&&(ok))
   {if(nr%i==0)ok=0;
   else
    i=i+1;}
    if(ok)
    {s=0; nr1=nr;
     while(nr1!=0)
     {s=s+nr1%10;nr1=nr1/10;}
     if(s<=m){cout<<nr<<endl;x=x+1;}
     };
     nr=nr+1;
   }
     getch();}
```

16. Să se afişeze toate numerele prime de 3 cifre care citite invers sunt tot numere prime.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
main()
{int i,j,ok,ok1,nr,ninv,n=100;
for(i=n;i<=999;i++)
   {j=2;
   ok=1;
   while(j<=sqrt(i)&&(ok))
   {if(i%j==0)ok=0;
```

```
else
 j=j+1;}
 if(ok)
  {//cout<<"numarul prim este="<<i<<endl;
  nr=i;ninv=0;
     while(nr)
      {ninv=ninv*10+nr%10;
       nr=nr/10 ;}//cout<<"inversul este"<<ninv<<endl;
     j=2;
    ok1=1;
    while(j<=sqrt(ninv)&&(ok1))
    {if(ninv%j==0)ok1=0;
     else j=j+1;}
     if(ok1){cout<<"nr.si nr.invers sunt prime"<<endl;
            cout<<i<<"  "<<ninv<<endl;}
     }
  }
  getch();
  }
```

17. Fie X un vector cu n componente numere întregi. Să se calculeze expresia: e=x1-x2+x3-x4+….±xn.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
main()
{int a[100],e[100];
int i,n;
cout<<"introduceti n=";
cin>>n;
for(i=1;i<=n;i++)
{cout<<"a["<<i<<"]=";
 cin>>a[i];
 e[i]=pow(-1,i+1)*a[i];}
 for(i=1;i<=n;i++)
 cout<<"a["<<i<<"]="<<a[i]<<" ";
 cout<<endl;
 for(i=1;i<=n;i++)
 cout<<"e["<<i<<"]="<<e[i]<<" ";
  getch();}
```

18. Se dă fişierul text Fractii.in care conține pe fiecare linie câte 2 numere naturale nenule mai mici sau egale cu 35000 separate printr-un spațiu cu semnificația că primul este numărătorul, iar al doilea este numitorul unei fracții. Să se creeze fişierul text Fractii.out care conține numărătorii şi numitorii fractiilor ireductibile corespunzătoare fracțiilor din Fractii.in

```
#include<fstream.h>
main()
{fstream f("fractii.in",ios::in);
fstream g("fractii.out",ios::out);
int i,j,x,y;

while(f>>i>>j)
{x=i;y=j;
while(x!=y)
if(x>y)x=x-y;
else y=y-x;
if(x==1)g<<i<<" "<<j<<endl;}
f.close();
g.close();}
```

19. Se introduc de la tastatură numere întregi formate din minim două cifre până la întâlnirea numărului 0. Să se afişeze câte numere au cifra unităților mai mică decât cifra zecilor.

```
#include<iostream.h>
#include<conio.h>
main()
{int n,y,z,x;
cout<<"introduceti un nr.din minim 2 cifre"<<endl;
cin>>n; x=0;
while(n)
{y=n%10;
 n=n/10;

 z=n%10;
 if(y>z)x=x+1;
 cin>>n;}
 cout<<"s-au introdus "<<x<<" numere care au cifra unitatilor mai mica decat cifra zecilor";
 getch();}
```

20. Fie un polinom P de gradul n cu coeficienți întregi. Să se scrie un program care să afişeze valoarea polinomului într-un punct x dat. Nu se vor utiliza vectori.

```
#include<iostream.h>
#include<conio.h>

main()
{int p,n,x,a,i;
cout<<"introduceti gradul polinomului"<<endl;
cin>>n;
cout<<"introduceti valoarea lui x"<<endl;
cin>>x;
cout<<"introduceti coeficientul" <<endl;
cout<<"a[0]=";
cin>>a;
p=a;
for(i=1;i<=n;i++)
```

```
{cout<<"introduceti coeficientul "<<endl;
cout<<"a["<<i<<"]=";
cin>>a;
p=p+a*x;
x=x*x;}

cout<<"valoarea polinomului este="<<p;
getch();}
```

21. Stabiliți dacă printre numerele întregi scrise într-un fişier text există sau nu cel puțin un număr par. În fişierul de intrare PAR.in se află numere cuprinse între 0 şi 1000000000. În fişierul Par.out se va scrie mesajul DA dacă există cel puțin un număr par şi NU în caz contrar.

```
#include<fstream.h>
main()
{fstream f("par.in",ios::in);
 fstream g("par.out",ios::out);
 int ok,i;
 while(f>>i&& ok)
 if(i%2==0)
 {g<<"da";ok=0;}
 if(ok)g<<"nu";
 f.close();
g.close();}
```

22. Să se determine primele n perechi de numere prietene. Două numere sunt prietene dacă fiecare este egal cu suma divizorilor celuilalt număr.

```
#include<iostream.h>
#include<conio.h>
main()
{int n,i,j,s,s1,k,x;
 cout<<"introduceti o valoare pentru n=";
 cin>>n;
 i=1;j=1;
 while(i<=n)
 {
  s=0;

  if(j==1)s=s+1;
  else
    if(j==2)s=s+3;
    else
     if(j==3)s=s+4;
   else s=s+1+j;
  for(k=2;k<=j/2;k++)
   if(j%k==0)s=s+k;
   cout<<"nr.este="<<j<<endl;
   cout<<"suma divizorilor este="<<s<<endl;
  x=j+1;
```

```
  if(s==x)
     {i=i+1;
  cout<<"nr.prietene sunt" <<endl;
  cout<<j<<" "<<x<<endl;
  }
  j=j+1;
  }
  getch();}
```

23. Să se determine numărul vârfurilor care au gradul minim într-un graf neorientat cu n vârfuri şi m muchii. Valorile n şi m se găsesc pe prima linie în fişierul mat.txt, iar pe următoarele m linii se găsesc perechi de numere reprezentând extremitățile muchiilor.

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
int a[50][50],n,i,j,m,s[50];
void citire (char nume[20], int a[50][50],int& n)
{ fstream f (nume,ios::in);
  int i,j;
  f>>n>>m;
  while (f>>i>>j) a[i][j]=a[j][i]=1;
  f.close();}
main()
{citire("min.txt",a,n);
int i=1,j,s=0,min,k;k=1;
while(i<=n)
{min=s;s=0;
for(j=1;j<=n;j++)s=s+a[i][j];
if(min>s){min=s;k=1;}
if(min==s) k=k+1;
i++;}
for (i=1;i<=n;i++)
 {for (j=1;j<=n;j++)
 cout<<a[i][j]<<"";
 cout<<endl;}
 cout<<"gradul min este="<<min<<endl;
 cout<<"nr.vf.cu grad min.este="<<k<<endl;
 getch();}
```

24. Să se creeze o listă simplu înlănțuită cu n noduri, fiecare nod din lista conține informații despre numele, media şi clasa unui elev. Să se afiseze toți elevii care au media cel puțin 7.

```
#include<iostream.h>
#include<conio.h>
struct Nod
{char nume[20];
 float medie;
```

```
 char clasa;
Nod*adr_urm;};
int n,i;
Nod *v;
void adaug(Nod*&v)
{Nod *c=new Nod;
cout<<"numele este"<<endl;
cin>>c->nume;
cout<<"media este"<<endl;
cin>>c->medie;
cout<<"clasa este"<<endl;
cin>>c->clasa;
c->adr_urm=v;
v=c;};
void tiparesc(Nod*v)
{Nod *c=v;
while (c)
if(c->medie>=7)
{cout<<c->nume<<" ";
cout<<c->medie<<" ";
cout<<c->clasa<<endl;
c=c->adr_urm;}
}
main()
{cout<<"numarul de elevi este"<<" ";
cin>>n;
for(int i=1;i<=n;i++)
adaug(v);
cout<<"elevii care au media cel putin 7 sunt:"<<endl;
tiparesc(v);
getch();
```

25. Să se verifice dacă un graf neorientat este conex.

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
int a[50][50],n,i,j,s[50];
void citire (char nume[20], int a[50][50],int& n)
{ fstream f (nume,ios::in);
  int i,j;
  f>>n;
  while (f>>i>>j) a[i][j]=a[j][i]=1;
  f.close();}
 void df(int nod)
  {int k;
  s[nod]=1;
  for(k=1;k<=n;k++)
```

```
 if((a[nod][k]==1)&& (s[k]==0))df(k);}
int conex()
 {int gasit=0,i;
 df(1);
 for(i=1;i<=n;i++)
 if(s[i]==0)gasit=1;
 return!gasit;}
main()
 {citire("gr.txt",a,n);
 for (i=1;i<=n;i++)
{for (j=1;j<=n;j++)
cout<<a[i][j]<<"";
cout<<endl;}
 if (conex())cout<<"graful este conex" ;
 else cout<<"graful nu este conex";
 getch();}
```

26. Să se afişeze toate numerele întregi, mai mici sau egale cu x, a căror sumă a cifrelor este impară.

```
#include<iostream.h>
#include<conio.h>
void main()
{int x,s,i,n;
cout<<"introduceti un numar x="<<endl;
cin>>x;
 for(i=1;i<=x;i++)
{s=0;n=i;
while(n)
{s=s+n%10;
n=n/10;}
if(s%2!=0)cout<<"suma cifrelor numarului "<<i<<" este impara"<<endl;}
getch();}
```

27. Se citeşte un număr natural n şi valorile k şi c. Să se insereze cifra c pe poziția k în numărul n.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{int n,inv,ninv,k,k1,x,poz,final,c,nr,b;
 cout<<"introduceti numarul n=";
 cin>>n;
 cout<<"introduceti pozitia k=";
 cin>>k;
 cout<<"introduceti cifra c=";
 cin>>c;
 nr=0;

 k1=0;inv=0;ninv=0;
 b=n;
while(b)
{ nr=nr+1;
b=b/10;}
poz=nr-k;
x=n/(pow(10,nr-k));
while(k1<poz)
{inv=inv*10+n%10;
k1=k1+1;n=n/10;}
while(inv)
```

```
{ninv=ninv*10+inv%10;
inv=inv/10;}
final=x*pow(10,nr-k+1)+c*pow(10,poz)+ninv;
cout<<"numarul obtinut este ="<<final;
getch();}
```

28. Fie vectorul x cu n componente numere întregi. Să se calculeze expresia E=x[1]-x[2] +x[3]-….±x[n]

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
main()
{int a[100],e[100];
int i,n,s=0;
cout<<"introduceti n=";
cin>>n;
for(i=1;i<=n;i++)
{cout<<"a["<<i<<"]=";
 cin>>a[i];
 e[i]=pow(-1,i+1)*a[i];
  s=s+e[i];}
 for(i=1;i<=n;i++)
 cout<<"a["<<i<<"]="<<a[i]<<" ";
 cout<<endl;
 for(i=1;i<=n;i++)
 {cout<<"e["<<i<<"]="<<e[i]<<" ";
 cout<<endl;
}
 cout<<"suma este:"<<s;
 getch();}
```

29. Să se construiască un vector v cu primii termeni ai şirului lui Fibonacci. Şirul are primii doi termeni egali cu 1 şi fiecare din termenii următori este egal cu suma dintre termenul precedent şi termenul anteprecedent.

```
#include<iostream.h>
#include<conio.h>
void main ()
{int i, v[100],n,a1,a2,a3;
cout<<"introduceti numarul n de termeni ai sirului ";
cin>>n;
v[1]=a1=1;
v[2]=a2=1;
for(i=3;i<=n;i++)
```

```
{a3=a2+a1;
v[i]=a3;
a1=a2;
a2=a3;}
cout<<"sirul lui Fibonacci este"<<endl;
for(j=1;i<=n;i++)
cout<<"v["<<i<<"]="<<v[i]<<" ";
getch();}
```

30. Fie v un şir de n elemente numere întregi. Să se construiască un vector u cu suma cifrelor elementelor cu proprietatea că sunt perfecte.

```
#include <iostream.h>
#include<conio.h>
void main()
{ //numere perfecte 6,28,496
int s, n,v[100],k,u[100];
cout << "Introduceti dimensiunea vectorului n= ";
cin >> n;
for(int i=1;i<=n;i++)
{cout<<"v["<<i<<"]=";
 cin>>v[i];}
 k=1;
 for(int i=1;i<=n;i++)
 { s=0;
 for (int j=1; j<=(v[i]/2); j++)
 if ((v[i]%j)==0) s=s+j;
 if (s==v[i]) {u[k]=s;k++;}}
 for(int i=1;i<=k-1;i++)
 cout<<"u["<<i<<"]="<<u[i]<<" ";
 getch();}
```

31. Fie a o matrice pătratică de ordinul n cu elemente numere întregi. Să se verifice dacă matricea este superior triunghiulară. O matrice este superior triunghiulară dacă toate elementele aflate sub diagonala principală sunt nule.

```
#include<iostream.h>
#include<conio.h>
void main()
{int i, j, a[10][10],n,gasit;
cout<<"introduceti dimensiunea n a matricei ";
cin>>n;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";
cin>>a[i][j];}
```

```
gasit=0;
for(i=1;i<=n;i++)
for(j=1;j<i;j++)
if(a[i][j]!=0)gasit =1;
if(!gasit)cout<<"matricea este superior triunghiulara";
else cout<<"matricea nu este superior triunghiulara";
getch();}
```

32. Se dă un şir x cu n elemente numere întregi. Să se determine cmmdc(a,b), unde a este cel mai mic şi b este cel mai mare element din şirul x.

```
#include<iostream.h>
#include<conio.h>
void main()
{int n, x[10],a,b,r;
cout<<"introduceti dimensiunea vectorului n=";
cin>>n;
for( int i=1; i<=n;i++)
{cout<<"x["<<i<<"]=";
cin>>x[i];}
a=b=x[1];
for( int i=2; i<=n;i++)
 {if(a>x[i])a=x[i];
  if(b<x[i]) b=x[i];}
 while(b!=0)
 {r=a%b;
 a=b;
 b=r;}
 cout<<"cmmdc intre a si b este "<<a;
 getch();}
```

33. Se dă un număr n întreg .Să se verifice dacă suma cifrelor lui n este număr prim.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{int n,s=0,x=1,i=2;
cout<<"introduceti numarul n="; cin>>n;
 while(n!=0)
 {s=s+n%10;
 n=n/10;}
 while(i<=sqrt(s)&& x)
 {if(s%i==0) x=0;
 else i=i+1;}
```

```
if(x) cout<<"suma cifrelor lui n este numar prim";
else  cout<<"suma cifrelor lui n nu este un numar prim";
getch();}
```

34. Se consideră un vector de maxim 10 elemente reale. Se cere sa se afişeze media aritmetică a valorilor strict pozitive din vector, cu două zecimale.

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
void main()
{float v[10],s,m;
 int i,n,p ;
 cout<<"introduceti o valoare pentru n=";
 cin>>n;
 for(i=1;i<=n;i++)
 {cout<<"v["<<i<<"]=";cin>>v[i];}
 s=0;p=0;
 for(i=1;i<=n;i++)
 if(v[i]>0){s=s+v[i];
 p=p+1;}
 m=s/p;
 cout<<"media aritmetica a numerelor pozitive
este:"<<setw(5)<<setprecision(3)<<setiosflags(ios::showpoint)<<m;
 getch();}
```

35. Se citeşte un vector de numere întregi. Să se memoreze în al doilea vector suma cifrelor pentru fiecare element din primul vector .

```
#include<iostream.h>
#include<conio.h>
void main()
{int v[10],i,n, a[10],s;
cout<<"dim.vect.este"; cin>>n;
cout<<"elementele vectorului sunt"<<endl;

for(i=1;i<=n;i++)
{cout<<"v["<<i<<"]="; cin>>v[i];}

for(i=1;i<=n;i++)
{s=0;
while(v[i]!=0)
{s=s+v[i]%10;
 v[i]=v[i]/10;}
 a[i]=s;}
for(int i=1;i<=n; i++)
cout<<"a["<<i<<"]="<<a[i]<<" ";
```

```
getch();}
```

36. Se citeşte o matrice cu n linii şi m coloane. Să se memoreze într-un vector sumele elementelor de pe fiecare linie a matricei.

```
#include<iostream.h>
#include<conio.h>
void main()
{int a[10][10],i,j,n,m,v[10],s;
cout<<"nr. de linii este"; cin>>n;
cout<<"nr. de coloane este"; cin>>m;
cout<<"elementele matricii sunt"<<endl;
for( i=1;i<=n;i++)
for ( j=1;j<=m;j++)
{cout<<"a["<<i<<"]["<<j<<"]="; cin>>a[i][j];}
for(i=1;i<=n;i++)
{s=0;
for(j=1;j<=m;j++)
 s=s+a[i][j];
 v[i]=s;}
 for(i=1;i<=n;i++)
 cout<<"v["<<i<<"]="<<v[i]<<" ";
getch();}
```

37. Se citesc 2 numere naturale m şi n. Calculati n^m .

```
#include<iostream.h>
#include<conio.h>
void main()
{int n, m, p,i;
cout<<"nr. n este=";
cin>>n;
cout<<"nr. m este=";
cin>>m;
p=1;
for(i=1; i<=m;i++)
p=p*n;
cout<<"n la puterea m este="<<p;
getch();}
```

38. Calculați suma: S=1*3+2*5+3*7+...+n*(2n+1).

```
#include<iostream.h>
#include<conio.h>
void main()
{int n, i, s;
```

```
cout<<"introduceti o valoare pentru n=";
cin>>n;
s=0;
for( i=1;i<=n;i++)
s=s+i*(2*i+1);

cout<<"suma este="<<s;
getch();}
```

39. Calculați suma: S=1+1*2+1*2*3+...+1*2*3*...*n.

```
#include<iostream.h>
#include<conio.h>
void main()
{int s, i, n,p;
cout<<"introduceti o valoare pentru n=";

cin>>n;
s=0; p=1;
for(i=1;i<=n;i++)
{p=p*i;
s=s+p;}
cout<<"suma este ="<<s;
getch();}
```

40. Se citesc mai multe numere până când introducem numărul 0. Să se afle:
a) câte sunt pozitive şi câte sunt negative

b) media aritmetică a numerelor impare.

```
#include<iostream.h>
#include<conio.h>
void main()
{int p,n,s,i;
 int x,m,y;
 p=0;n=0;s=0;i=0;
 cout<<"citirea se face pana la introducerea valorii 0"<<endl;
 cout<<"introduceti o valoare=";
 cin>>x;
 while(x!=0)
 {

 if (x>0) p=p+1;
 else n=n+1;
 y=x%2;
  if(y!=0)
  {s=s+x;
   i=i+1;}
 cout<<"introduceti o valoare=";
```

```
cin>>x;}
cout<<"s-au introdus un numar de "<<p<<" numere pozitive"<<endl;
cout<<"s-au introdus un numar de "<<n<<" numere negative"<<endl;
m=s/i;
cout<<"media numerelor impare este="<<m<<endl;

getch();}
```

41. Se citesc mai multe numere până când introducem numărul 0. Să se calculeze suma celor pare şi produsul celor impare.

```
#include<iostream.h>
#include<conio.h>
void main()
{int p,s,x;
 p=1;s=0;

 cout<<"citirea se face pana la introducerea valorii 0"<<endl;
 cout<<"introduceti o valoare=";
 cin>>x;
 while(x!=0)
 {
 if (x%2==0) s=s+x;
 else p=p*x;
 cout<<"introduceti o valoare=";
 cin>>x;}
 cout<<"suma numerelor pare este "<<s<<endl;
  cout<<"produsul numerelor impare este "<<p<<endl;
  getch();}
```

42. Se citeşte un număr natural n. Să se verifice dacă este palindrom(un număr este palindrom dacă, citit de la stânga la dreapta şi de la dreapta la stânga, are aceeaşi valoare; de exemplu, 121).

```
#include<iostream.h>
#include<conio.h>
void main()
{int n,inv,y;
cout<<"introduceti un numar"<<endl;
cin>>n;
y=n; inv=0;
while(n)
{inv=inv*10+n%10;
n=n/10;}
if(y==inv)cout<<"numarul introdus este palindrom";
else cout<<"numarul introdus nu este palindrom";
getch();}
```

43. Să se determine toți divizorii primi ai unui număr natural dat.

```
#include<iostream.h>
#include<conio.h>

void main ()
{ int n,i;
cout<<"introduceti o valoare pentru n=";
cin>>n;
i=2;
while(n!=1)
{if(n%i==0)
{cout<<i<<endl;
 while(n%i==0)n=n/i;}
 i=i+1;}
 getch();}
```

44. Scrieți un program Pascal/C++ care citeşte de la tastatură un cuvânt format din cel mult 20 de caractere, doar litere mici ale alfabetului englez. Programul determină transformarea cuvântului citit prin înlocuirea fiecărei vocale a cuvântului cu litera mare corespunzătoare, restul literelor nemodificându-se. Programul afişează pe ecran cuvântul obținut. Se consideră vocale literele din mulțimea {a,e,i,o,u}.

Exemplu: **pentru cuvântul anotimp se va afişa AnOtImp.**

```
#include<iostream.h>
#include<string.h>
#include<ctype.h>
#include<conio.h>
void main()
{ char s[21], v[]="aeiou";
int i, n;
cin.get(s,21);
n=strlen(s);
for(i=0;i<n;i++)
if(strchr(v,s[i]))
s[i]=toupper(s[i]);
cout<<s;
getch();}
```

45. Se consideră un text cu maximum 255 de caractere în care cuvintele sunt separate prin unul sau mai multe spații. Primul caracter din textul citit este o literă, iar cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un Program Pascal/C++ care citeşte de la tastatură textul şi îl transformă, înlocuind prima literă a fiecărui cuvânt cu litera mare corespunzătoare, restul caracterelor rămânând nemodificate. Textul astfel transformat va fi afişat pe ecran.

Exemplu**: dacă de la tastatură se introduce textul: afară este frig se va afişa pe ecran: Afară Este Frig**

```
#include<iostream.h>
#include<string.h>
#include<ctype.h>
#include<conio.h>
char s[255];
int i,n;
void main()
{ cin.get (s, 255);
n=strlen(s);
s[0]=toupper(s[0]);
for (i=1;i<strlen(s);i++)
```

```
if(s[i-1]==' '&& s[i]!=' ')
s[i]=toupper(s[i]);
cout<<s;
getch();}
```

46. Scrieți un program Pascal/C++ care citeşte de la tastatură un număr natural n(1≤i≤n) şi construieşte în memorie o matrice cu n linii şi n coloane, numerotate de la 1 la n, astfel încât fiecare element situat pe o linie i(1≤i≤n) şi pe o coloană (1≤j≤n) va fi egal cu suma dintre i şi j. Programul va afişa matricea pe ecran, câte o linie a matricei pe o linie a ecranului, elementele de pe aceeaşi linie fiind separate prin câte un spațiu.

***Exemplu*: dacă n este egal cu 4, se va afişa matricea:**

2 3 4 5
3 4 5 6
4 5 6 7
5 6 7 8

```
#include<iostream.h>
#include<conio.h>
int n, i, j, a[24][24];
void main()
{cin>>n;
for(i=1;i<=n;i++)
 for(j=1;j<=n;j++)
  a[i][j]=i+j;

for(i=1;i<=n;i++)
 for(j=1;j<=n;j++)
cout<<a[i][j]<<" ";
cout<<endl;}
getch();}
```

47. Se consideră un şir de n numere naturale. Să se afişeze fracția subunitară ireductibilă care se poate forma din numărul minim şi numărul maxim din şir.

***Exemplu*: pentru n egal cu 4 şi numerele 3 2 5 8 se va afişa ¼**

```
#include<iostream.h>
#include<conio.h>
long i,min,max,x,y,n;
void main()
{ cin>>n>>min; max=min;
for(i=1; i<n; i++)
{ cin>>x;
 if(x<min) min=x;
if(x> max) max=x;
}
```

```
x=min; y=max;
while (x!=y)
  if (x>y) x=x-y;
  else y=y-x;
cout<<min/y<<"/"<<max/y;

getch();}
```

48. Se citeste de la tastatură un vers al unei poezii si o silabă .Să se realizeze un program care determină numărul de aparitii al silabei citite în textul respectiv.

***Exemplu*:Pentru versul „Un curcubeu multicolor" si silaba"cu"se va afisa 2**

```
#include<iostream.h>
#include<stdio.h>
#include<string.h>
#include<conio.h>
char vers[256],s[256];
int nr,p,l,n;
void  main()  {
gets(vers); gets(s); nr=0;
while(strstr (vers,s)!=NULL){
p=strstr(vers,s)-vers;
l=strlen(s);
strcpy(vers+p,vers+p+1);
nr++;
          }
cout<<nr;
getch();
}
```

49. Se consideră două cuvinte formate din literele mari si mici ale alfabetului englez. Verificati dacă ele sunt anagrame. Două siruri sunt anagrame,dacă unul dintre ele este format din caracterele celuilalt, eventual într-o altă ordine.

***Exemplu*"are","era".**

```
#include<iostream.h>

#include<string.h>
#include<conio.h>
char x[21], y[21],*p;
void main()
{cin>>x>>y;
 if ( strlen(x)==strlen(y))
{ while (strchr(y,x[0]!=null && x[0]!=0))
    {p=strchr(y,x[0]);
      strcpy(p,p+1);
      strcpy(x,x+1);
      }
```

```
if ( x[0]==0 && y[0]==0)
   cout<<"da";
else
    cout<<"nu";
}
else cout<<"nu";
getch();
}
```

50. Să se realizeze un program pentru rezolvarea ecuației de gradul I: ax+b=0, a,b aparțin R.

```
#include<iostream.h>
#include<conio.h>
void main()
{ float a,b,x;
cout<<"introduceti coeficientii ec. de gr.I"<<endl;
cin>>a>>b;
if(a==0)
 if(b==0)
 cout<<"o infinitate de solutii"<<endl;
 else cout<<"nu are solutii"<<endl;
 else {x=-b/a; cout<<" radacina "<<x;}
getch();}
```

51. Să se realizeze un program pentru rezolvarea ecuației de gradul II: ax2+bx+c=0, a,b,c aparțin R, a $\neq$ 0.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{ float a,b,c,x1,x2,d;
cout<<"introduceti coeficientii ec. de gr.II"<<endl;
cin>>a>>b>>c;
if(a==0){cout<<"ecuatie de gradul I"<<endl;
 if(b==0)
    if(c==0) cout<<"o infinitate de solutii"<<endl;
    else cout<<"nu are solutii"<<endl;
 else {x1=-c/b; cout<<"cu radacina "<<x1;}}
else
{d=b*b-4*a*c;
 if(d>0){x1=(-b+sqrt(d))/(2*a);x2=(-b-sqrt(d))/(2*a);
        cout<<"ec. are 2 radacini reale diferite"<<endl;
        cout<<"x1="<<x1<<" "<<"x2="<<x2;}
 else if(d==0) {x1=-b/(2*a);cout<<"ec.are 2 solutii reale identice"<<endl;
               cout<<"x1=x2="<<x1;}
```

```
      else cout<<"ec. nu are solutii reale"<<endl;}
 getch();}
```

52. Se citeşte un număr de la tastatură, să se afişeze oglinditul său (numărul obținut prin inversarea cifrelor numărului inițial).

```
#include<iostream.h>
#include<conio.h>
void main()
{int n, inv=0;
cout<<"introduceti numarul n=";
cin>>n;
while(n!=0)
{inv=inv*10+n%10;
n=n/10;}
cout<<"oglinditul numarului este "<<inv;
getch();}
```

53. Se citeşte un număr întreg n. Să se elimine cifrele impare şi să se afişeze numărul obținut. Exemplu : dacă se citeşte 8325 atunci se va afişa 82.

```
#include<iostream.h>
#include<conio.h>
void main()
{int n,v,inv;
 cout<<"introduceti numarul n=";
 cin>>n;
 v=0;inv=0;
while(n)
{if(n%2==0)
v=v*10+n%10;
n=n/10;
}
while(v)
{inv=inv*10+v%10;
v=v/10;}
cout<<"numarul format numai din cifrele pare este "<<inv;
getch();}
```

54. Se citeşte un număr întreg n de la tastatură. Să se înlocuiască cifrele impare cu cifra 0. Exemplu: dacă se citeşte 8325 atunci se va afişa 8020.

```
#include<iostream.h>
#include<conio.h>
void main()
```

```
{int n,inv,ninv,k,k1;
 cout<<"introduceti numarul n=";
 cin>>n;
 k=0;k1=0;inv=0;ninv=0;
while(n)
{if(n%2==0)
inv=inv*10+n%10;
else inv=inv*10;
n=n/10;k=k+1;}
while(inv)
{ninv=ninv*10+inv%10;
inv=inv/10;k1=k1+1;}
if(k==k1)
cout<<"numarul format este "<<ninv;
else
cout<<"numarul format este"<<ninv*10;
getch();}
```

55. Să se afle cifra de control a unui număr întreg n. Pentru aflarea cifrei de control se calculează suma cifrelor numărului respectiv şi se obține un număr. Dacă numărul este format din mai multe cifre se calculează din nou suma cifrelor sale, etc. Daca suma nou obținută este formată dintr-o singură cifră, acea cifra va fi cifra de control a numărului inițial.

```
#include<iostream.h>
#include<conio.h>

void main()
{
  int n, a, b, s;
  cout<<"n="; cin>>n;
  a=n;
  s=n;
  while(s>9)
{
    s=0;
    while(n>0) {
      s=s+n%10;
      n=n/10;
    }
    n=s;
  }
  cout<<"Cifra de control a lui "<<a<<" este "<<s;
  getch();
}
```

56. Să se afiseze al n-elea termen al şirului lui Fibonacci, folosind o funcție recursivă. irul are primii 2 termeni egali cu 1 şi fiecare din următorii termeni este egal cu suma dintre termenul precedent şi termenul anteprecedent.

```
#include<iostream.h>
#include<conio.h>
int n;
int fib(int n)
{if(!n) return 0;
else if(n==1)return 1;
else return fib(n-1)+fib(n-2);}
main()
{cout<<"n="; cin>>n;
cout<<fib(n);
getch();}
```

57. Să se calculeze combinarile de n elemente luate câte k, folosind o funcție recursivă, conform relației de recurență: $C^k_n = C^k_{n-1} + C^{k-1}_{n-1}$, cu cazurile limită $C^0_n = C^n_n = 1$.

```
#include<iostream.h>
#include<conio.h>
int n, k;
int comb(int n,int k)
{if (k==0) return 1;
else if(k==n) return 1;
else return comb(n-1,k)+comb(n-1,k-1);}
main()
{cout<<"n=";cin>>n;
cout<<"k=";cin>>k;
cout<<comb(n,k);
getch();}
```

58. Se dă un număr natural n. Să se creeze un sir care să conțină divizorii săi în ordine descrescătoare. Să se introducă în şir media aritmetică a divizorilor astfel încât şirul să rămână ordonat descrescător. În cazul în care aceasta nu este număr întreg atunci se va rotunji la partea sa întreagă.

```
#include<iostream.h>
#include<conio.h>
void main()
{int n,v[100],u[100],j,k,s,m,i,gasit;
cout<<"introduceti nr. n=";
```

```
cin>>n;
v[1]=n; j=2;
for(int i=n/2;i>=2;i--)
if(n%i==0)
{ v[j]=i;
j++;} k=j;
v[k]=1;s=0;
for(int i=1; i<=k;i++)
{cout<<v[i]<<" "; s=s+v[i];}
cout<<endl;
m=s/k;
cout<<"media divizorilor este :"<<m<<endl;
i=1; gasit=1;
while(i<=k && gasit)
{ i++;
if(v[i]>m && v[i+1]<m) gasit=0;}
 s=i; j=1;
for(i=1;i<=s;i++)
 {u[j]=v[i];j++;}
 u[j]=m;
 for(i=s+1;i<=k;i++)
   {u[j+1]=v[i];j++;} k++ ;
 for(i=1;i<=k;i++ )
 cout<<u[i]<<" ";
getch();}
```

59. Se consideră mulțimea {1,2,...,n}. Se cer toate submulțimile acestei mulțimi .

```
#include<iostream.h>
#include<conio.h>
int n, sol[10],i;
void back(int k)
{if(k==n+1)
 {for (i=1;i<=n;i++)
 //cout<<sol[i];
 if(sol[i]!=0) cout<<i<<" ";

 cout<<endl;}
{sol[k]=-1;
while(sol[k]<1)
{sol[k]++;
back(k+1);}}}
void main()
{ cout<<"n=";cin>>n;
back(1);
getch();}
```

60. Să se determine numărul total de descompuneri ale unui număr natural, care nu pot fi obținute una din alta prin permutarea termenilor.

Ex:

4=1+1+1+1

4=1+1+2

4=1+3

4=2+2

4=4

```
#include<iostream.h>
#include<conio.h>
int sol[100],n,i,s;
void back(int k)
{if(s==n)
{for(i=1;i<=k-1;i++)
cout<<sol[i];
cout<<endl;}
else
{ if(k>1) sol[k]=sol[k-1];
else sol[k]=0;
while(sol[k]+s<n)
{sol[k]++;
s+=sol[k];
back(k+1);
s-=sol[k];}}}
main()
{cout<<"n=";cin>>n;
back(1);
getch();}
```

61. Un cal şi un rege se află pe o tablă de şah. Unele câmpuri sunt „arse”, pozițiile lor fiind cunoscute. Calul nu poate păşi pe câmpurile „arse”, iar câmpul de pe care pleacă calul (cu excepția câmpului de start şi a celui în care se află regele) „arde”. Se cere să se determine o succesiune de mişcări ale calului (dacă ea există), care deplasează calul din poziția inițială în poziția în care se află regele, apoi înapoi în poziția inițială.

Intrare: fişierul text atilla.in conține pe prima linie dimensiunea tablei de şah n (n<15)

Linia a doua conține coordonatele câmpului de start a calului

Linia a treia conține coordonatele regelui

Următoarele linii conțin coordonatele câmpurilor „arse”.

Ieşire: La ecran se afişează tabla de şah cu indicatoarele numărului mutării şi a poziției calului după mutarea respectivă.

```
#include<fstream.h>
#include<stdlib.h>
#include<conio.h>
ifstream f("atilla.in");
```

```
int dx[]={-2,-1,1,2,2,1,-1,-2};
int dy[]={1,2,2,1,-1,-2,-2,-1};
int t[100][100],st[100][2],i,j,n,xc,yc,xr,yr,k,lin,col;
int solutie(int k)
{ int ok=0;
  for(j=1;j<=k;j++)
    if(st[j][0]==xr&&st[j][1]==yr)
      ok=1;
  if(k>1&&st[k][0]==xc&&st[k][1]==yc&&ok==1)
    return 1;
  else return 0;
}
void tipar(int k)
{ cout<<endl;
  for(i=1;i<=k-1;i++)
    cout<<i<<" "<<st[i][0]<<" "<<st[i][1]<<endl;
    getch();
  exit(EXIT_SUCCESS);
}
void back(int k,int lin,int col)
{  int i,linie,coloana;
  if(solutie(k-1))tipar(k);
  else{
    st[k][0]=lin;
    st[k][1]=col;
    for(i=0;i<=7;i++)
    { linie=lin+dx[i];
      coloana=col+dy[i];
      if(linie<=n&&linie>=1&&coloana<=n&&coloana>=1&&t[linie][coloana]==0)
      {t[linie][coloana]=1;
       back(k+1,linie,coloana);
       t[linie][coloana]=0;
      }
    }
  }
}
void main()

{ f>>n;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
      f>>t[i][j];
  f>>xc>>yc;
  f>>xr>>yr;
  f.close();
  back(1,xc,yc);
}
```

62. Se citeşte un număr întreg de la tastatură. Scrieti o funcție care calculează numărul de cifre ale lui n. Folosind această funcție stabiliți dacă n se găseşte în intervalul [100,10000).(Nu se va compara numarul n cu 100 şi nici cu 10000).

```
#include<iostream.h>
#include<conio.h>
int numar (int n)
{int x=0;
while(n!=0)
{x=x+1;
n=n/10;
}
return x;}
void main()
{int n;
cout<<"numarul este=";cin>>n;
if (numar (n)>=3 && numar (n)<5)
cout<<"Numarul apartine intervalului  [100,10000)";
else cout<< "Numarul nu apartine intervalului  [100,10000)" ;
getch();}
```

63. Fie un graf neorientat. Să se determine dacă graful conține cicluri .

```
#include<iostream.h>

#include<conio.h>
#include<fstream.h>

int s[50],a[50][50],gasit,n;
void citire (char nume_fis[20], int a[50][50], int&n)
{int i,j;
fstream f (nume_fis, ios::in);
f>>n;
while(f>>i>>j)
a[i][j]=a[j][i]=1;
f.close();}

void df(int nod)
{int k;
s[nod]=1;
for (k=1;k<=n;k++)
 if (a[nod][k]==1)
 {
 a[k][nod]=0;
 if(s[k]==0) df(k);
 else gasit=1;}}
 main()
```

```
{citire("graf.txt",a,n);
df(1);
if(gasit) cout<<" contine cicluri";
else cout<<"nu contine cicluri";
getch();}
```

64. Folosind metoda divide et impera: să se numere elementele pare ale unui vector cu n numere întregi .

```
#include<iostream.h>
#include<conio.h>
 int a[20],b[20],n;
 void citire()//datele de intrare

 {
 cout<<"n=";cin>>n;
 for(int  i=1;i<=n;i++)
 {cout<<"b["<<i<<"]=";cin>>b[i];
 }}
 int par(int p,int u)
 {int m,s1,s2;
 if(p==u)
    if(b[p]%2==0)return 1;
    else return 0;
       else

       {m=(p+u)/2;
        s1=par(p,m);
        s2=par(m+1,u);
        return s1+s2;
       }
 }
 void main()
 {
 citire();
 cout<<"nr.elementelor pare este="<<par(1,n)<<endl;
 getch();
 }
```

65. Fiind dată o hartă cu n țări, se cer toate modalitățile de colorare a hărții, utilizând cel mult m culori, astfel încât două țări cu frontiera comună să fie colorate diferit.

```
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
#include<math.h>
```

```
int st[20],n,k,a[10][10],i,j,m;


int valid(int k)
{for(int i=1;i<k;i++)
if(st[k]==st[i] && a[i][k]==1) return 0;
return 1;}

void back(int k)
{int i;
if(k==n+1)
 {for(int i=1;i<=n;i++) cout<<"tara numarul:"<<i<<" culoarea:"<<st[i]<<endl;
 getch();
 exit(EXIT_SUCCESS);
}
else
for(i=1;i<=m;i++)
{st[k]=i;
if(valid(k))
back(k+1);}}

main()
{cout<<"numarul de tari=";cin>>n;
cout<<"numarul de culori, maxim 4=";cin>>m;
for(int i=1;i<=n;i++)
for(int j=1;j<=i-1;j++)
{ cout<<"a["<<i<<","<<j<<"]=";
 cin>>a[i][j];
 a[j][i]=a[i][j];}
back(1);
}
```

66. Se citeşte de la tastatură matricea de adiacență a unui graf orientat G = (X, U) cu n noduri şi m muchii. Să se determine numărul vârfurilor pentru care gradul interior este strict mai mare decât gradul exterior.

```
#include<iostream.h>
#include<conio.h>

void main()
{int n, a[50][50],i,j,d1[50],d2[50],nr;
cout<<"numarul de noduri este=";cin>>n ;
cout<<"introduceti matricea de adiacenta"<<endl;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";
```

```
cin>>a[i][j];
cout<<endl;}
nr=0;
 for(i=1;i<=n;i++)
 for(j=1;j<=n;j++)
 d1[i]=d1[i]+a[i][j];
 for(j=1;j<=n;j++)
  for(i=1;i<=n;i++)
 d2[j]=d2[j]+a[i][j];
 for(i=1;i<=n;i++)
 {  cout<<"d1["<<i<<"]="<<d1[i]<<endl;
    cout<<"d2["<<i<<"]="<<d2[i]<<endl;
 if(d1[i]<d2[i])nr=nr+1;}
 cout<<"nr.de noduri la care gradul int.este mai mare decat gradul exterior este="<<nr;
getch();}
```

67. Să se afiseze toate numerele prime de trei cifre care citite invers, sunt tot numere prime.
***Ex*. Un astfel de număr este 761 (761 este prim si 167 este tot prim).**

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{int i,gasit,ninv,nr,nr1;
  long n;
for(n=100;n<=999;n++)
 {i=2;gasit=1;nr=n;nr1=n;ninv=0;
  while(i<=sqrt(n) && gasit)
  if(n%i==0) gasit=0;
  else i++;
   if(gasit)
    while(nr1!=0)
   {ninv=ninv*10+nr1%10;
    nr1=nr1/10;}
if(nr && ninv && gasit)
cout<<nr<<endl; }
getch();}
```

68. Se spune că n este deosebit dacă există un număr natural m astfel încât n=m+S(m), unde S(m) este suma cifrelor lui m. Să se scrie un program care verifică dacă un număr natural n dat este deosebit.
***Ex*. 1235 este deosebit (1235=1225+10).**

```
#include<iostream.h>
#include<conio.h>
void main()
{int n, nr, s,x;
```

```
cout<<"numarul este=";
cin>>n;
s=0;
nr=n;
while(nr)
{s=s+nr%10;
nr=nr/10;}
x=n-s;
if(x>=0)
{cout<<"numarul este deosebit"<<endl;
cout<<n<<"="<<s<<"+"<<x;}
else
cout<<"numarul nu este deosebit";
getch();}
```

69. O carte se paginează cu numere de la 1 la n (20<=n<=10000, n natural). Fiind dat n, să se afle câte cifre au fost folosite la paginare.

```
#include<iostream.h>
#include<conio.h>

void main()
{int n,i,j,nr;
nr=0;
cout<<"introduceti nr. de pagini=";
cin>>n;

for(i=1;i<=n;i++)
{j=i;
while(j)
{j=j/10;
nr=nr+1;}}
cout<<"cifrele folosite la paginare sunt="<<nr;
getch();}
```

70. Să se creeze o listă liniară simplu înlănțuită cu n numere întregi citite de la tastatură. Să se afişeze numărul cheilor negative, pozitive şi nule din listă.

```
#include<iostream.h>
#include<conio.h>
struct nod
{int info;
nod* adr;};
nod* v;nod* sf;
int i;
void adaugare(nod*&v, nod*&sf, int val)
```

```
{nod* c;
if(v==0)
{v=new(nod);
v->info=val;
v->adr=0;
sf=v;}
else
{c=new (nod);
sf->adr=c;
c->info=val;
c->adr=0;
sf=c;}}
void listare(nod*v)
{nod*c=v;
while(c)
{cout<<c->info<<" ";
c=c->adr;}
cout<<endl;}

main()
{ int n,neg=0,poz=0,nr=0,p;
cout<<"introduceti o val. ptr. n=";
cin>>n;
for(i=1;i<=n;i++)
{cin>>p;
adaugare(v,sf,p);};
listare(v);
nod*c=v;
while(c)
{if(c->info<0) neg++ ;
else if(c->info>0) poz++;
else nr++;
c=c->adr;}
cout<<"nr.negative="<<neg<<endl;
cout<<"nr.pozitive="<<poz<<endl;
cout<<"nr.nule="<<nr<<endl;
getch();}
```

71. Să se determine numărul componentelor pare aflate pe poziții impare într-o listă cu componente numere întregi.

***Ex*. Pentru o carte cu 100 de pagini se folosesc 192 de cifre.**

```
#include<iostream.h>
#include<conio.h>
struct nod
{int info;
nod* adr;};
```

```
nod* v;nod* sf;
int i;
void adaugare(nod*&v, nod*&sf, int val)
{nod* c;
if(v==0)
{v=new(nod);
v->info=val;
v->adr=0;
sf=v;}
else
{c=new (nod);
sf->adr=c;
c->info=val;
c->adr=0;
sf=c;}}
void listare(nod*v)
{nod*c=v;
while(c)
{cout<<c->info<<" ";
c=c->adr;}
cout<<endl;}

main()
{ int n,nr=0,p;
cout<<"introduceti o val. ptr. n=";
cin>>n;
for(i=1;i<=n;i++)
{cin>>p;
adaugare(v,sf,p);};
listare(v);
nod*c=v;
if (c->info%2==0)nr=nr+1;
do
{
c=c->adr->adr;
if (c->info%2==0)nr=nr+1;
}
while(c->adr!=0);
cout<<" nr.pare aflate pe pozitiile impare sunt="<<nr<<endl;
getch();}
```

72. Fiind dată o listă liniară simplu înlănțuită, să se scrie un program care utilizând funcții, inserează un nod în interiorul listei şi şterge un nod de la începutul listei .

```
#include<iostream.h>
#include<conio.h>
struct nod
```

```
{int info;
nod* adr;};

nod* v;nod* sf;
int i;

void adaugare(nod*&v, nod*&sf, int val)
{nod* c;
if(v==0)
{v=new(nod);
v->info=val;
v->adr=0;
sf=v;}
else
{c=new (nod);
sf->adr=c;
c->info=val;
c->adr=0;
sf=c;}}

void listare(nod*v)
{nod*c=v;
while(c)
{cout<<c->info<<" ";
c=c->adr;}
cout<<endl;}

void inserare_dupa(nod* v,nod*& sf, int val, int val1)
{nod*  c=v;nod *d;
while(c->info!=val)
c=c->adr;
d=new (nod);
d->info=val1;d->adr=c->adr;
c->adr=d;
if(d->adr==0)sf=d;}

void sterg(nod* &v)

{nod*man;
man=v;
v=v->adr;
delete man;}
main()
{ int n,p,val,val1;
cout<<"introduceti o val. ptr. n=";
cin>>n;
for(i=1;i<=n;i++)
{cin>>p;
adaugare(v,sf,p);};
```

```
listare(v);
cout<<"dupa ce val. se insereaza?";
cin>>val;
cout<<"ce val.se insereaza?";
cin>>val1;
inserare_dupa(v,sf,val,val1);
listare(v);
cout<<"acum sterg prima inregistrare"<<endl;
sterg(v);
listare(v);
getch();}
```

73. Să se creeze o listă circulară cu n numere întregi citite de la tastatură şi să se afişeze elementele ei.

```
#include<iostream.h>
#include<conio.h>
struct nod
{int info;
nod* adr;};

nod* v;nod* sf;
int i;

void adaugare(nod*&v, nod*&sf, int val)
{nod* c;
if(v==0)
{v=new(nod);
v->info=val;
v->adr=0;
sf=v;}
else
{c=new (nod);
sf->adr=c;
c->info=val;
c->adr=v;

sf=c;}}
void listare(nod*v)
{nod*c=v;cout<<"lista circulara este:"<<endl;
do
{cout<<c->info<<" ";
c=c->adr;}
while(c!=v);
cout<<endl;}

main()
```

```
{ int n,p;
cout<<"introduceti o val. ptr. n=";
cin>>n;
for(i=1;i<=n;i++)
{cout<<"introduceti o valoare=";cin>>p;cout<<endl;
adaugare(v,sf,p);};
listare(v);
getch();}
```

74. Scrieți un program care citeşte de la tastatură un număr natural x, de tip longint şi creează o listă liniară simplu înlănțuită cu cifrele acestui număr. Fiecare nod al listei va conține exact o cifră din x.

```
#include<iostream.h>
#include<conio.h>
struct nod
{int info;
nod* adr;};

nod* v;nod* sf;
int i;

void adaugare(nod*&v, nod*&sf, int val)
{nod* c;
if(v==0)

{v=new(nod);
v->info=val;
v->adr=0;
sf=v;}
else
{c=new (nod);
c->info=val;
c->adr=v;
v=c;
}}


void listare(nod*v)
{nod*c=v;
cout<<"lista formata este:"<<endl;
while(c)
{cout<<c->info<<" ";
c=c->adr;}
cout<<endl;}


main()
{ long int x;
```

```
int p;
cout<<"introduceti o val. ptr. x=";
cin>>x;

while(x!=0)

{p=x%10;
adaugare(v,sf,p);
x=x/10;};
listare(v);
getch();}
```

75. Fiind dată o listă liniară dublu înlănțuită, să se determine maximul cheilor din listă.

```
 #include<iostream.h>

#include<conio.h>
struct Nod
    {int info;
      Nod *next,*back;
      };

Nod *prim, *ultim;
void creare_lista()
{Nod *c;
 c=new Nod;
 cout<<"info ";

 cin>>c->info;
 if(!prim)
   {prim=c;
    prim->next=0;
    prim->back=0;
    ultim=prim; }
 else
   {ultim->next=c;
    c->back=ultim;
    ultim=c;
    ultim->next=0;

   }
}

void listare_stanga_dreapta()
{Nod *c;
 c=prim;
 while(c)
    {cout<<c->info<<" ";
     c=c->next;}
}
```

```
void listare_dreapta_stanga()
{Nod *c;
 c=ultim;
 while(c)
    {cout<<c->info<<" ";
     c=c->back;}
}
int maxim()
{Nod *c;
int max;
 c=prim;max=c->info;
 while(c)
    {if(c->info>max)max=c->info;
     c=c->next;}
     return max;
}

void main()
   {int n,i;
    clrscr();
    cout<<"cate elemente va avea lista?";
    cin>>n;
    for(i=1;i<=n;i++)
      creare_lista();
    cout<<endl<<"Elementele listei de la stanga la dreapta sunt:"<<endl;
    listare_stanga_dreapta();
    cout<<endl<<"Elementele listei de la dreapta la stanga sunt:"<<endl;
    listare_dreapta_stanga(); cout<<endl;
    cout<<"valoarea maxima din lista este:"<<maxim();
    getch();}
```

76. Fiind dat un număr natural n să se determine toate modalitățile de descompunere a lui în sume de termeni naturali.

n – se introduce de la tastatură
rezultatele se afişează în fişierul part.out

Exemplu:

n=5
5=1+1+1+1+1
5=1+1+1+2
5=1+1+2+1
5=1+1+3
5=1+2+1+1
5=1+2+2
5=1+3+1
5=1+4
5=2+1+1+1
5=2+1+2

5=2+2+1
5=2+3
5=3+1+1
5=3+2
5=4+1

5=5

```
#include<iostream.h>
#include<fstream.h>
int sol[100],n,i,s;
void back(int k)
{if(s==n)
{ fstream f("part.txt",ios::app);
for(i=1;i<=k-1;i++)
f<<sol[i];
f<<endl;}
else
{
sol[k]=0;
while(sol[k]+s<n)
{sol[k]++;
s+=sol[k];
back(k+1);
s-=sol[k];}}}
main()
{cout<<"n=";cin>>n;
back(1);
}
```

77. Scrieți un program care citeşte de la tastatură două numere naturale n şi m (2≤m≤10, 2≤n≤10) şi care construieşte în memorie şi apoi afişează o matrice A cu n linii (numerotate de la 1 la n) şi m coloane (numerotate de la 1 la m) cu proprietatea că fiecare element Aij memorează cea mai mare dintre valorile indicilor i si j (1≤i≤n, 1≤j≤m). Matricea se va afişa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

```
#include<iostream.h>

#include<iomanip.h>
int n,m,i,j,a[10][10];
void main()
{cout<<"introduceti o val. ptr.n=";
cin>>n;
cout<<"introduceti o val. ptr.m=";
cin>>m;
for(i=1;i<=n;i++)
 for(j=1;j<=m;j++)
  if(i>j)a[i][j]=i;
  else a[i][j]=j;
```

```
for(i=1;i<=n;i++)
 {for(j=1;j<=m;j++)
cout<<a[i][j]<<" ";
cout<<endl;}
getch();}
```

78. Se citesc din fişierul "mat1.in" , de pe prima linie, două numere naturale n1 şi m1 şi de pe următoarele n1 linii o matrice cu n1 linii şi m1 coloane. Analog, se citesc din fişierul "mat12in" , de pe prima linie, două numere naturale n2 şi m2 şi de pe următoarele n2 linii o matrice cu n2 linii şi m2 coloane. Să se scrie un program în care se calculează produsul celor două matrici. Dacă produsul se poate face atunci se va afişa matricea rezultată în fişierul "mat.out", iar în caz contrar se va afişa un mesaj corespunzător.

```
#include <fstream.h>
#include <conio.h>
ofstream g("mat.txt");
ifstream f("mat1.txt");
ifstream h("mat2.txt");

int main()
{
int a[50][50],b[50][50],c[50][60],n1,m1,n2,m2,sum,i,j,k;
f>>n1>>m1;
h>>n2>>m2;
if(m1!=n2) g<<"Inmultirea nu este posibila!"<<endl;
else{
for(i=1;i<=n1;i++)
for(j=1;j<=m1;j++) f>>a[i][j];
for(i=1;i<=n2;i++)
for(j=1;j<=m2;j++) h>>b[i][j];
cout<<endl;
for(i=1;i<=n1;i++)
for(j=1;j<=m2;j++)
{sum=0;
for(k=1;k<=n2;k++){sum+=(a[i][k]*b[k][j]);
                        c[i][j]=sum;

}                       }
for(i=1;i<=n1;i++) {
for(j=1;j<=m2;j++)  g<<c[i][j]<<' ';
g<<endl;
}}
return 0;
}
```

79. Se citesc de la tastatură două cuvinte. Să se scrie un program care verifică dacă unul este anagrama celuilalt. Un cuvânt este anagramă pentru altul dacă cele două cuvinte conțin aceleaşi litere de acelaşi număr de ori, dar nu neapărat în aceeaşi ordine.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

void sort(char *s)
{
char i,j,aux[30],cc;
strcpy(aux,s);
for (i=0;i<strlen(s)-1;i++)
  for (j=i+1;j<strlen(s);j++)
   if (aux[i]<aux[j])
    {
     cc=aux[i];
     aux[i]=aux[j];
     aux[j]=cc;
    }
strcpy(s,aux);
return;
}

void main()
{
char s1[30],s2[30];

clrscr();
printf("Cuv1:");scanf("%s",s1);sort(s1);
printf("Cuv2:");scanf("%s",s2);sort(s2);

if (strcmp(s1,s2)==0) printf("Sunt anagrame\n"); else
            printf("Nu sunt anagrame\n");
getch();
}
```

80. Se citesc de pe prima linie a fişierului "monede.in" două numere s şi n şi de pe următoarea linie se citeşte un şir de n numere naturale. Se cere să se afişeze în fişierul "monede.out" toate posibilitățile de plată a sumei s folosind n tipuri de monede ale căror valori sunt memorate în şirul citit.

```
 #include<iostream.h>
#include<fstream.h>
int sol[100], a[100],n,i,s,suma;
void back(int k)
```

```
{if(s==suma)
 {fstream g("monede1.txt",ios::app);
 g<<"solutie"<<endl;
 for(i=1;i<=k-1;i++)
 if(sol[i])
 g<<sol[i]<<"monede de"<<a[i]<<endl;
 g<<endl;
g.close();}
 else
 {sol[k]=-1;
 while(sol[k]*a[k]+s<suma && k<n+1)
 {sol[k]++;
 s+=sol[k]*a[k];
 back(k+1);
 s-=sol[k]*a[k];}}}
 main()
 { fstream f("monede.txt",ios::in);
f>>suma;
f>>n;
 for(i=1;i<=n;i++)
 f>>a[i];
f.close();
 back(1);}
```

81. Se citesc din fişierul "pol.in", de pe prima linie, gradele a două polinoame. De pe următoarele următoarele două linii se citesc două şiruri de numere reale care reprezintă coeficienţii celor două polinoame, în ordinea crescătoare a gradelor. Să se scrie un program în care se calculează produsul polinoamelor. Rezultatul se va afişa în fişierul "pol.out".

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
struct nod
{float coef;
int grad;
nod* adr;};

nod* v;nod* sf;
int i;

void adaug(nod*&v, nod*&sf, int gr,float cf)
{nod* c;
c=new (nod);
c->grad=gr;
c->coef=cf;
c->adr=0;
if(v==0) v=sf=c;
else
```

```
{sf->adr=c;
sf=c;}}

void sterg(nod*v)
{nod*c=v;
while(v)
{v=v->adr;
delete c;
c=v;}}

void afis(nod*v)
{int i=0;
while(v)
{if(i) cout<<"+";
cout<<v->coef<<"x**"<<v->grad;
v=v->adr;
i++;
}
cout<<endl<<endl;}

nod* adun(nod* v1,nod*v2)
{nod* c1=v1;
nod* c2=v2;
nod* v=0,*sf;
while(c1&&c2)
if(c1->grad==c2->grad)
 {if(c1->coef+c2->coef)
 adaug(v,sf,c1->grad,c1->coef+c2->coef);
 c1=c1->adr;
 c2=c2->adr;
 }
 else
    if(c1->grad>c2->grad)
    {adaug(v,sf,c1->grad,c1->coef);
     c1=c1->adr;}

    else

   {adaug(v,sf,c2->grad,c2->coef);
    c2=c2->adr;}
if(c1)
while(c1)
{adaug(v,sf,c1->grad,c1->coef);
c1=c1->adr;}
else
  while(c2)
{adaug(v,sf,c2->grad,c2->coef);
c2=c2->adr; }
return v;}
```

```
nod *mulmonom(nod *v, int gr, float cf)
{nod *vf=0; *sf;
while(v)
{adaug(vf,sf,v->grad+gr,v->coef*cf);
v=v->adr;
}
return vf;}

nod *mul(nod* v1, nod*v2)
{nod*v=0;
while(v2)
{nod *vman=0, *vman1;
vman=mulmonom(v1,v2->grad,v2->coef);
vman1=v;
v=adun(v,vman);
sterg(vman);
sterg(vman1);
v2=v2->adr;
}
return v;}

main()
{
char linie[100];
nod*p1,*p2,*vf=0,*sf;
int gr1,gr2,gr;
float cf;
fstream f1("pol1.txt",ios::in);
f1>>gr1;
cout<<gr1<<endl;
gr=gr1;
while(f1.getline(linie,100,'\n'))
{f1>>cf; cout<<cf<<" "<<endl;
adaug(vf,sf,gr,cf);
gr=gr-1;}cout<<gr<<endl;

f1.close();
p1=vf;
afis(p1); vf=0 ;
fstream f2("pol2.txt",ios::in);
 f2>>gr2;
cout<<gr2<<endl;
gr=gr2;
while(f2.getline(linie,100,'\n'))
{f2>>cf;
adaug(vf,sf,gr,cf);
gr=gr-1;}
```

```
p2=vf;
afis(p2);
f2.close();

nod*p=mul(p1,p2);

fstream f3("pol3.txt",ios::out);
int i=0, v=p;
while(v)
{if(i) f3<<"+";
f3<<v->coef<<"x**"<<v->grad;
v=v->adr;
i++;}
f3.close();
afis(p);
getch();}
```

82. Se consideră o matrice A cu m linii şi n coloane ce contine numere întregi. Să se scrie un program care afişează, pe o singură linie în fişierul text A.txt, elementele situate pe conturul matricei. Se porneşte din poziția (1,1) şi parcurgerea se va face în sensul acelor de ceasornic.

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{int n,m,i,j,a[100][100];
 cout<<"n="; cin>>n;
 cout<<"m="; cin>>m;
 for(i=1;i<=n;i++)
  for(j=1;j<=m;j++)
  cin>>a[i][j];
  for(i=1;i<=n;i++)
 {  for(j=1;j<=m;j++)
  cout<<a[i][j];
  cout<<endl;}
  fstream f("A.txt",ios::out);
  for(j=1;j<=m;j++)
  f<<a[1][j];
  for(i=2;i<=n;i++)
  f<<a[i][m];
  for(j=m-1;j>=1;j--)
  f<<a[n][j];
  for(i=n-1;i>=2;i--)
  f<<a[i][1];
  f.close();
  getch();}
```

83. Se consideră o matrice A cu m linii şi n coloane de numere reale şi un vector V cu m elemente (numere reale). Să se verifice dacă elementele vectorului V formează o coloana a matricei A. În caz afirmativ să se afişeze în fişierul verific.txt numărul coloanei respective; în caz contrar să se afişeze în fişier 'NU'.

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{int n, m, i,j,gasit, v[100], a[100][100];
cout<<"m=";cin>>m;
cout<<"n=";cin>>n;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";
```

```
cin>>a[i][j];}
for(i=1;i<=m;i++)
{cout<<"v["<<i<<"]=";
cin>>v[i];}
fstream f("verific.txt", ios::out);

for(j=1;j<=n;j++)
  {i=1;gasit=1;
   while((i<=m) && (gasit))

if(a[i][j]==v[i])i++ ;
else gasit=0;
if(gasit) f<<j;}
 if(gasit==0) f<<"NU";
}
```

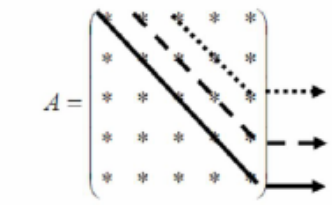
84. Se consideră o matrice A cu m linii şi n coloane ce conține numere reale. Să se şteargă din matrice linia a 3-a. Să se afişeze în fişierul A.txt matricea obținută în format corespunzător. Se consideră că matricea are cel puțin 3 linii(m>= 3).

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{int n, m, i,j, a[100][100],b[100][100];
fstream f("A.txt",ios::out);
cout<<"m=";cin>>m;
cout<<"n=";cin>>n;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";
cin>>a[i][j];}
for(i=1;i<=2;i++)
for(j=1;j<=n;j++)
b[i][j]=a[i][j];
for(i=4;i<=m;i++)
for(j=1;j<=n;j++)
b[i-1][j]=a[i][j];

for(i=1;i<=m;i++)
{ for(j=1;j<=n;j++)
  f<<a[i][j];
 f<<endl;}
 f<<endl;
  for(i=1;i<=m-1;i++)
 { for(j=1;j<=n;j++)
 f<<b[i][j];
  f<<endl;}
  }
```

85. Se consideră o matrice A cu m linii şi n coloane de numere întregi. Să se introducă o nouă linie la începutul matricii. Să se afişeze matricea obținută în format corespunzător în fişierul A.txt.

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{int n, m, i,j, a[100][100];
fstream f("A.txt",ios::out);
cout<<"m=";cin>>m;
cout<<"n=";cin>>n;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";
cin>>a[i+1][j];}

for(i=1;i<=m;i++)
 { for(j=1;j<=n;j++)
  f<<a[i+1][j];
 f<<endl;}
 f<<endl;
for(j=1;j<=n;j++)
{cout<<"a["<<1<<"]["<<j<<"]=";
cin>>a[1][j];}
  for(i=1;i<=m+1;i++)
 { for(j=1;j<=n;j++)
 f<<a[i][j];
  f<<endl;}
  f.close();
  }
```

86. Se consideră o matrice pătratică de dimensiune n(n>2) ce conține numere întregi. Să se calculeze media aritmetică a elementelor situate pe a doua paralelă cu diagonala principală situată deasupra diagonalei principale.Valoarea se va afişa în fisierul B.txt.



A doua paralelă cu diagonala principală

Prima paralelă cu diagonala principală

Diagonala principală

```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{int n,nr=0,s=0, i,j, a[100][100];
```

```
float m;
fstream f("B.txt",ios::out);
cout<<"n=";cin>>n;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";
cin>>a[i][j];}
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if(i==j){s=s+a[i][j+2];nr=nr+1;}
m=s/(nr-2);
for(i=1;i<=n;i++)
 { for(j=1;j<=n;j++)
  f<<a[i][j];
 f<<endl;}
 f<<endl;
 f<<m;
  f.close();
  }
```

87. Fie şirul următor definit prin recurență:

$$a_n = \begin{cases} -n + a_{n-3}, daca\ n\ este\ impar; \\ 2n + a_{n-3}, daca\ n\ este\ par, \end{cases}$$

$$a_0 = 0,$$

$$a_1 = 1,$$

$$a_2 = 2.$$

Sa se scrie un subprogram recursiv care calculează termenii acestui şir şi să se folosească acest subprogram pentru a calcula al 10-lea termen al şirului. Dimensiunea vectorului şi elementele acestuia se găsesc pe o linie în fişierul A.TXT.

```
#include<conio.h>
#include<fstream.h>
int functie(int n)
{if(n==0)return 0;
 else if(n==1)return 1;
     else if(n==2) return 2;
          else  if(n%2==0)
          return 2*n+functie(n-3) ;
          else return functie (n-3)-n;}


void main()
{
fstream f("A.txt",ios::out);
f<<functie(10);
f.close();}
```

88. Să se scrie un subprogram recursiv care afişează elementele situate pe poziţii pare într-un vector ce conţine numere întregi. Elementele vor fi afişate fiecare pe câte un rând şi vor fi afişate în ordinea în care apar în vector. Dimensiunea vectorului şi elementele acestuia se găsesc pe o linie în fişierul C.TXT

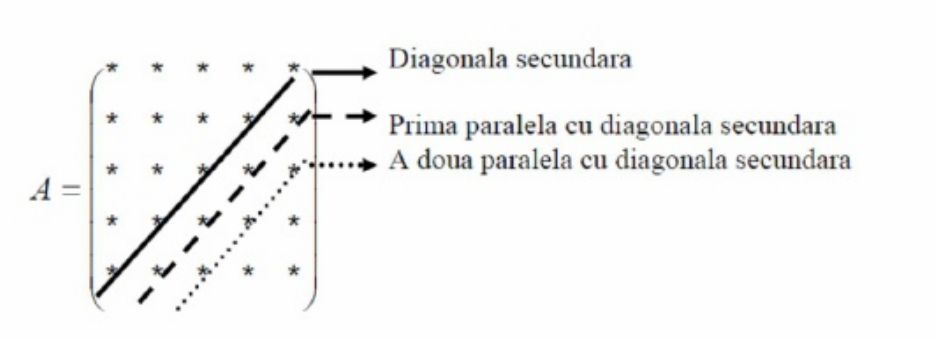
```
#include<iostream.h>
#include <fstream.h>
#include <conio.h>

ifstream f("C.txt");
void afis(int n, int v[100])
{
if(n)
{
afis(n-2,v);
cout<< v[n];
cout<<endl;}

}
void main()
{ int n,i, v[100];
f>>n;
for(i=1;i<=n;i++)
f>>v[i];
if(n%2!=0)n=n-1;
afis(n,v);
getch();}
```

89. Fie A o matrice pătratică de dimensiune n(n>2) de numere reale. Cele doua diagonale împart matricea în 4 regiuni(4 triunghiuri). Să se verifice dacă suma elementelor din interiorul triunghiului de sus este egală cu suma elementelor din interiorul triunghiului de jos. Dimensiunea matricei şi elementele acesteia se găsesc în fisierul INTRARE.txt astfel: pe prima linie se găseşte n, pe următoarele n linii se găseste câte o linie din matrice.

```
#include<iostream.h>
#include <fstream.h>

#include <conio.h>
ifstream f("INTRARE.txt");
void main()
{ int n,i,j, a[100][100],s1,s2;
f>>n;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
f>>a[i][j];
for(i=1;i<=n;i++)
{for(j=1;j<=n;j++)
```

```
cout<<a[i][j]<<" ";
cout<<endl;}
s1=0;s2=0;
for(i=1;i<=(n)/2-1;i++)
for(j=i+1;j<n+1-i;j++)

s1=s1+a[i][j];
cout<<"suma elementelor din triunghiul de sus este="<<s1<<endl;
for(i=(n)/2+2;i<=n;i++)
for(j=n-i+2;j<i;j++)
s2=s2+a[i][j];
cout<<"suma elementelor din triunghiul de jos este="<<s2<<endl;
if(s1==s2) cout<<"cele doua sume sunt egale"<<endl;
else  cout<<"cele doua sume nu sunt egale"<<endl;
getch();}
```

90. Se consideră o matrice pătratică de dimensiune n(n>2) ce conține numere întregi. Să se afișeze în fișierul A.TXT suma elementelor situate pe prima paralelă cu diagonala secundară situată sub diagonala secundară.




```
#include<iostream.h>
#include<conio.h>
#include<fstream.h>
void main()
{int n,s=0, i,j, a[100][100];
fstream f("A.txt",ios::out);
cout<<"n=";cin>>n;
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";

cin>>a[i][j];}
for(i=1;i<=n;i++)
for(j=1;j<=n;j++)
if(i+j==n+1)s=s+a[i+1][j];

for(i=1;i<=n;i++)
 { for(j=1;j<=n;j++)
  f<<a[i][j];
 f<<endl;}
 f<<endl;
 f<<s;
```

```
f.close();
}
```

91. Se consideră o matrice cu m linii şi n coloane de numere întregi. Dimensiunile matricei şi elementele ei se găsesc în fişierul A.TXT astfel: pe prima linie se găsesc valorile m si n. Pe următoarele m linii se găsesc elementele celor m linii ale matricei. Să se verifice dacă există în matrice două linii identice şi în caz afirmativ să se afişeze numerele de ordine ale celor două linii, iar în caz contrar să se afişeze mesajul 'NU'.

```
#include<iostream.h>
#include<conio.h>
void main()
{int m,n,i,j,gasit,k,a[100][100];

cout<<"m=";cin>>m;
cout<<"n=";cin>>n;
for(i=1;i<=m;i++)
for(j=1;j<=n;j++)
{cout<<"a["<<i<<"]["<<j<<"]=";
cin>>a[i][j];}
 for(i=1;i<=m;i++)
  for(k=i+1;k<=m ;k++)
  {gasit=1;
  for(j=1;j<=n && gasit;j++)
 if(a[i][j]==a[k][j])gasit=1 ;
 else gasit=0;
 if (gasit) cout<<"liniile "<<k<<" "<<i<<"sunt identice"<<endl;
 else cout<<"liniile"<<k<<" "<<i<<"nu sunt identice"<<endl;}
 getch();}
```

92. Scrieți programul care citeşte din fisierul text BAC.TXT un număr natural n (n<100) şi un şir cu n numere întregi din intervalul [100 ;999] ; programul construieşte un sir de numere rezultat prin înlocuirea fiecărui număr din şirul citit cu numărul obținut prin interschimbarea cifrei unitatilor cu cifra sutelor. Numerele din noul şir se vor afişa in fisierul text Rez.TXT separate printr-un singur spațiu. De exemplu , pentru n=3 si şirul 123 , 904 , 500 , se afişează 321 , 409 , 5.

```
#include<iostream.h>
#include <fstream.h>
#include <conio.h>
ofstream g("rez.txt");
ifstream f("bac.txt");

void main()
{int n,i,o,v[20],c;
f>>n;
for(i=1;i<=n;i++)
```

```
f>>v[i];
for(i=1;i<=n;i++)
{o=0;
c=v[i]%10;
o=o*10+c;
v[i]=v[i]/10;
c=v[i]%10;
o=o*10+c;
v[i]=v[i]/10;
o=o*10+v[i];
v[i]=o;
g<<v[i]<<endl;
}
}
```

93. Scrieți un program care afișează în fișierul text Rez.TXT toate numerele naturale formate din cifre identice , mai mari decât 10 şi mai mici decât o valoare dată n, citită din fișierul text BAC.TXT ,n<=2.000.000. De exemplu pentru n=195 , se afișează : 11 , 22 , 33 , 44 , 55 , 66, 77 , 88 , 99 , 111.

```
#include<iostream.h>
#include <fstream.h>
#include <conio.h>

ifstream f("bac.txt");
ofstream g("rez.txt");
int n,i;
int identic(int n)
{int d,c;
c=n%10;
while (n>0)
{d=n%10;
if(d==c)
n=n/10;
else
return 0;}
if(n==0)
return 1;}
void main()
{f>>n;
for (i=10;i<=n;i++)
{if(identic(i))
g<<i<<endl;}}
```

94. Scrieți un program C/C++ care citeşte din fisierul text BAC.TXT trei numere naturale x, y si k, (1<x<y<2000000, k<1000) şi afişează în fisierul text Rez.TXT k numere prime din intervalul [x, y]. Dacă nu există k numere prime în intervalul [x,y] se vor afişa toate numerele prime găsite, iar pe linia următoare se va afişa mesajul "s-au găsit mai puține numere prime: " urmat de numărul acestora. De exemplu, pentru x=3, y=12 si k=5 se vor afişa pe ecran: 3 5 7 11

s-au găsit mai puține numere prime:4

```
#include<iostream.h>
#include <fstream.h>
#include <conio.h>

ifstream f("bac.txt");
ofstream g("rez.txt");

void main()
{int x,y,k,n,d,a,p;
f>>x;
f>>y;
f>>k;
a=0;
n=x;
while ((n>=x)&&(n<=y)&&(a<k) )
{d=2;p=1;
while((d<=n/2)&&(p==1))
if(n%d==0)
p=0;
else
d++;
if(p)
{g<<n<<endl;a++;}
n++;}
if(a<k)
g<<"au fost gasite mai putine nr prime "<<a;}
```

95. Scrieți un program care citeşte din fişierul text BAC.TXT , cel mult 100 de numere naturale aflate pe o singura linie, formate din cel mult 9 cifre fiecare, separate prin spatii şi dintre acestea le afişează în fişierul text Rez.TXT doar pe acelea care au proprietatea de a fi palindrom. Dacă nu se găsesc numere palindrom, se va afisa pe ecran valoarea -1. Un număr are proprietatea de a fi palindrom daca citit de la dreapta la stanga sau de la stanga la dreapta are aceeaşi valoare. De exemplu 1221 este palindrom, în timp ce 1210 nu este palindrom. Exemplu : daca din fişierul BAC.TXT se citesc numerele : 7341 , 8228 , 660 , 2 , 80 , 131 , atunci pe ecran se vor afişa : 828 , 2 , 131.

```
#include<iostream.h>
#include <fstream.h>
#include <conio.h>
```

```
int palindrom(int n)
{int copie=n,oglindit,c;
oglindit=0;
while(n!=0)
{c=n%10;
oglindit=oglindit*10+c;
n=n/10;}
if (copie==oglindit)
return 1;
else return 0;}

void main()
{int n=0,p;
p=0;
fstream f("bac.txt",ios::in);
fstream g("rez.txt",ios::out);
while(!f.eof())
{f>>n;
if(palindrom(n))
{g<<n<<" ";
p=1;}}
if(p==0)
g<<"-1";
f.close();
g.close();
}
```

96. Pentru orice număr natural nenul n definim n factorial (notat n!) ca fiind produsul tuturor numerelor naturale nenule mai mici sau egale cu n (n! = 1*2*3*…*n) De exemplu : 3!=1*2*3=6 5!=1*2*3*4*5=120 . Scrieți un program care determina numărul de cifre nule aflate pe ultimele poziții consecutive ale valorii obținute in urma evaluării lui n!, n fiind un număr natural de cel mult 4 cifre citit din fişierul text BAC.TXT .De exemplu daca n=10, n!=3628800 rezultatul va fi 2 deoarece 3628800 are două zerouri la sfârşit.

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
void main()
{int n,nr,c,p,i;
 nr=0; p=1;
fstream f ("bac.txt", ios::in);
f>>n; cout<<n<<endl;
for(i=1;i<=n;i++)
p=p*i;cout<<p<<endl;
while(p)
{c=p%10;
if(c==0)
```

```
nr=nr+1;
p=p/10;}

cout<<"numarul de zerouri este="<<nr;
getch();}
```

97. Scrieți un program care citeşte din fişierul text DATE.IN cel mult 100 de numere naturale nenule aflate pe o singura linie, formate din cel mult 4 cifre fiecare, separate prin spatii şi scrie în fişierul DATE.OUT numerele, în ordinea inversă față de cea în care au fost citite, pe o singura linie separate prin spații. De exemplu dacă din fişierul DATE.IN se citesc numerele 93 207 15 1982 3762, atunci conținutul fişierului DATE.OUT va fi 3762 1982 15 207 93.

```
#include<fstream.h>
void main()
{int n,i=1,j,v[100];
fstream f("date.in",ios::in);
fstream g("date.out",ios::out);
while(f>>n)
{v[i]=n;i++;}
i=i-1;
for(j=i;j>0;j--)
g<<v[j]<<" ";
f.close();
g.close();}
```

98. Scrieți un program care citeşte de pe prima linie a fişierului text BAC.TXT trei numere naturale a, b, c formate din cel mult patru cifre fiecare, separate prin cate un spațiu şi afişează pe ecran cel mai mare divizor comun al acestora. De exemplu, dacă din fişier se citesc numerele : 9 27 15, atunci se afişează 3.

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
int cmmdc (int x,int y)
{while (x!=y)
if(x>y)x=x-y;
else
y=y-x;
return x;}
void main()
{int a,b,c,d;
fstream f("bac.txt",ios::in);
f>>a>>b>>c;
f.close();
d=cmmdc(a,b);
cout<<cmmdc(c,d);
getch();}
```

99. Se citeşte de la tastatura un număr natural nenul n (n<1000). Scrieți un program care construieşte fişierul text bac.txt care să conțină, pe prima linie, toți divizorii lui n în ordine strict descrescătoare. Divizorii vor fi separați prin spațiu. De exemplu, dacă n=10 , atunci fişierul bac.txt va conține : 10 , 5 , 2 , 1

```
#include<iostream.h>
#include<fstream.h>
void main()
{int d,n;
cout<<"n=";cin>>n;
fstream g("bac.txt",ios::out);
for(d=n;d>=1;d--)
if(n%d==0)
g<<d<<" ";
g.close();}
```

100. Se citeşte de la tastatura un număr natural nenul n care are cel mult 9 cifre. Să se afişeze în fişierul Date.out numărul k, natural, astfel încât produsul 1*2*…*(k-1)*k să aibă o valoare cât mai apropiată de numărul n. De exemplu, dacă se citeşte numărul n=25 fişierul Date.out are următorul conținut: 4. iar dacă se citeşte numărul n=119 fişierul Date.out are următorul conținut: 5

```
#include<iostream.h>
#include<fstream.h>

void main()
{int k,p,r,n;
cout<<"n=";cin>>n;
k=1;
fstream g("date.out",ios::out);
p=1;
while(p<=n)
{p=p*k;
k++;}
p=p/(k-1);
r=p*k;

if((n-p)<=(r-n))
g<<"numarul k este "<<k-2;
else
g<<"numalul k este "<<k-1;
g.close();
}
```

101. Scrieți un program care citeşte un număr natural n>1 cu maximum 9 cifre din fişierul DATE.IN, şi afişează în fişierul DATE.OUT valoarea celui mai mic divizor prim a lui n, precum şi puterea la care acest divizor apare în descompunerea în factori primi a numărului n.

```
#include<iostream.h>
#include<fstream.h>
int n,p,c;
void cmmd(int n)
{int gasit=0,d=2;
while((d<n/2)&&(gasit==0))
{if(n%d==0)
{c=d;
gasit=1;}
else
d++;}
p=0;
while(n%c==0)
{n=n/c;
p=p+1;}}
void main()
{ fstream f("date.in",ios::in);
fstream g("date.out",ios::out);
f>>n;
cmmd(n);
g<<c; g<<"la puterea";
g<<p;
f.close();
g.close();
}
```

102. Să se scrie un program care să calculeze şi să afişeze in fisierul DATE.OUT produsul cifrelor pare şi suma cifrelor impare a unui număr natural n citit din fişierul text BAC.TXT

```
#include<iostream.h>
#include<fstream.h>
void main()

{unsigned int n, s=0, p=1, c;
fstream f("bac.txt",ios::in);
fstream g ("date.out",ios::out);
f>>n;
while(n)
{
c=n%10;
if(c%2==0)
p=p*c;
else
s=s+c;
```

```
n=n/10;
}
g<< p <<" ";
g<< s;
f.close();

g.close();}
```

103. Să se verifice dacă două cuvinte citite din fişierul text BAC.TXT (cuvintele având maxim 50 de caractere fiecare) sunt rime, adică ultimele p caractere ale celor două cuvinte coincid, unde p (2<=p<=10) este dat.

```
#include<iostream.h>
#include<string.h>
#include<fstream.h>
#include<conio.h>
char s1[51], s2[51];
int p, x, y, ok=0;
void main()
{
fstream f("bac.txt",ios::in);
f>> s1;
f>>s2;
f>>p;
x=strlen(s1)-1;
y= strlen(s2)-1;
for( int i=1;i<=p && ok==0 ;i++)
{
if(s1[x]!=s2[y])
ok=1;
x--;
y--;
}
if(ok)
cout<<"Nu sunt rime";
else
cout<<"Sunt rime";
getch();

}
```

104. Scrieți un program care construieşte o matrice pătrată de ordin n formată din valorile 1 si 2 astfel încât elementele de pe diagonala principala si secundară sa fie egale cu 1, iar restul elementelor cu 2. Valoarea lui n se citeşte din fişierul text BAC.TXT: n număr natural (2<23), iar matricea se va afişa în fişierul text ies.txt, câte o linie a matricei pe fiecare rând cu spații între elementele fiecărei linii.

```
#include<iostream.h>
#include<fstream.h>
int a[10][10], i, j, n;
```

```
void main()
{ fstream f("bac.txt",ios::in);
fstream g("ies.txt",ios::out);
f>>n;
for(int i=1;i<=n;i++)
for(int j=1;j<=n;j++)
if(i==j || i+j==n+1)
a[i][j]=1;
else
a[i][j]=2;
for(i=1;i<=n;i++)
{
for(j=1;j<=n;j++)
g<< a[i][j];
g<< endl;
}
f.close();
g.close();
}
```

105. Scrieți un program care citește din fișierul text BAC.TXT un șir de cel mult 50 de caractere și construiește fișierul atestat.txt ce conține șirul și prefixele acestuia de lungime cel puțin 1, fiecare pe câte o linie, în ordinea descrescătoare a lungimii prefixelor.

```
#include<fstream.h>
#include<string.h>
#include<conio.h>
char s[51];
int n;
void main()
{
fstream f("bac.txt",ios::in);
fstream g("atestat.txt",ios::out);
f>>s;
n=strlen(s);
while(n>=0)
{
for(int i=0; i<n; i++)
g<< s[i];
g<< endl;
n--;
}
f.close();
g.close();
}
```

106. Să se scrie un program care să numere cuvintele dintr-un text citit din fişierul "cuvinte.txt". Textul conține cuvintele separate numai printr-un spațiu, fără semne de punctuație, singurul semn de punctuație este '.' de la sfârşitul textului. Numărul de cuvinte se va afişa pe ecran.

```
#include<fstream.h>
#include<string.h>
#include<conio.h>
void main()
{
ifstream f("cuvinte.txt");
char s[100];
int k=1,n;
f.getline(s,100);n=strlen(s);
for(int i=0;i< n;i++)
if(s[i]==' '&& s[i+1]!=' ')
k++;
cout<<"numarul de cuvinte din text este:"<< k;
f.close();
getch();
}
```

107. Scrieți programul care citeşte din fisierul text atestat.txt o valoare naturală n (2≤n≤100), construieşte în memorie şi apoi afişează în fişierul text ies.txt o matrice a cu n linii şi n coloane, numerotate de la 1 la n, care conține numerele naturale, în ordine crescătoare, de la 1 la n^2, dispuse pe coloane, în ordine crescătoare. Astfel coloana 1 va conține numerele de la 1 la n, coloana 2 numerele de la n+1 la 2*n, coloana 3 de la 2*n+1 la 3*n ş.a.m.d. Pe câte o linie a matricei pe câte o linie, elementele fiecărei linii fiind separate între ele prin câte un spațiu.
Exemplu: pentru n = 4 se va afişa matricea:
1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16

```
#include<iostream.h>
#include<fstream.h>

int a[10][10], k=1,s,n;
void main()
{
fstream f("atestat.txt",ios::in);
fstream g("ies.txt",ios::out);
f>>n;
for(int j=1;j<=n;j++)
for(int i=1;i<=n;i++)
{
a[i][j]=k;
k++;
```

```
}
for(int i=1;i<=n;i++)
{
for(int j=1;j<=n;j++)
g<< a[i][j]<<" ";
g<< endl;
}
f.close();
g.close();
}
```

108. Se dă n număr natural. Să se afişeze în fişierul text ies.txt cel mai apropiat număr prim față de n. Programul va conține cel puțin un subprogram.
Exemplu: n=24 se va afişa 23, pentru n=26 se va afişa 29

```
#include<fstream.h>
int prim(int x)
{
if(x==0 || x==1) return 0;
for(int i=2;i<=x/2;i++)
if(x%i==0)
return 0;
return 1;
}
void main()
{fstream f("ies.txt",ios::out);
unsigned n, x, y;
cout<< "n=";
cin>>n;
x=n+1;
y=n-1;
while(prim(x)==0) x++;
while(prim(y)==0) y--;
if(x-n>n-y)
f<< y;
else
f<< x;
}
```

109. Se consideră un vector cu n (1< n<100) componente numere naturale. Să se determine componentele vectorului cu cei mai mulți divizori şi să se afişeze în fişierul text ies.txt. Se va folosi un subprogram pentru citirea vectorului şi unul pentru calculul numărului de divizori ai unui număr.

```
#include<iostream.h>
#include<fstream.h>
int n, a[100];
void citire ()
{  int i;
for(i=1;i<=n;i++)
cin>>a[i];
}
int div(int x)
{   int i, nr=0;
for(i=2;i<=x;i++)
if(x%i==0)    nr++;
return nr;
}
void main()
{ fstream f("ies.txt",ios::out);
int i,max=0;
cin>>n;
citire();
for(i=1;i<=n;i++)
if(div(a[i])>max)
max=div(a[i]);
for(i=1;i<=n;i++)
if(div(a[i])==max)
f<< a[i]<<" ";
}
```

110. Fisierul text 'date.txt' conține un text scris cu litere mari pe una sau mai multe linii. Scrieți un program care să determine acea literă care apare de cele mai multe ori. Dacă există mai multe astfel de litere, se vor afişa toate.
Exemplu: Dacă fişierul are conținutul "VARA ESTE BINE AICI" literele afişate sunt a,e şi i.

```
#include<iostream.h>
#include<string.h>
#include<fstream.h>
#include<conio.h>
ifstream f("date.txt");
int n;
char a[100];
void citire()
{int i;
i=0;
f>>a[i];
while(!f.eof())
{cout<<a[i];
i++;
strupr(a);
f>>a[i];
```

```
}
n=i;
f.close();
}
void maxim()

{
int b[50];
cout<<"\n numarul de caractere="<<n<<"\n";
int k=0;
for(char c='A';c<='Z';c++){
            b[k]=0;
            for(int j=0;j<n;j++)
                        if(c==a[j]) b[k]++;

            k++;


}
int max=b[0];
for(int i=1;i<k;i++)
            if(max<b[i])max=b[i];
cout<<"caracterele cu numar maxim de aparitii="<<max<<endl;
for(int i=0;i<k;i++)
            if(b[i]==max)cout<<char(65+i)<<" ";

}


void  main()
{citire();
maxim();
getch();
}
```


http://slidepdf.com/reader/full/culegere-de-probleme-rezolvate-in-c-c 76/76